## Алгоритмы, печатающие числа, наименьшее значение вводимого числа

**1.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа a и b. Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 13.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a := 0; b := 0;      while x > 0 do          begin              a := a+1;              b := b+(x mod 100);              x := x div 100;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 100;

end;

Т. к. опе­ра­тор div воз­вра­ща­ет целую часть от де­ле­ния, то при де­ле­нии на 100 это рав­но­силь­но от­се­че­нию по­след­них двух цифр.

На каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся две по­след­них цифры до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0. Для того, чтобы a стало рав­ным 2, *x* долж­но быть трёхзнач­ным или четырёхзнач­ным.

Те­перь рас­смот­рим из­ме­не­ние b:

while x>0 do begin

b:=b+(x mod 100);

end;

Опе­ра­тор mod воз­вра­ща­ет оста­ток от де­ле­ния, при де­ле­нии на 100 это по­след­ние две цифры *x*. Разобьём 13 на два сла­га­е­мых так, чтобы можно было со­ста­вить трёхзнач­ное число: 13 = 1 +12. Ис­ко­мое число — 112.

Ответ: 112.

**2.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа a и b. Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 7.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a := 0; b := 1;      while x > 0 do          begin              a := a+1;              b := b\*(x mod 100);              x := x div 100;          end;      writeln(a); write(b); end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 100;

end;

Т. к. опе­ра­тор div воз­вра­ща­ет целую часть от де­ле­ния, то при де­ле­нии на 100 это рав­но­силь­но от­се­че­нию по­след­них двух цифр.

На каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся две по­след­них цифры до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0. Для того, чтобы a стало рав­ным 2, *x* долж­но быть трёхзнач­ным или четырёхзнач­ным.

Те­перь рас­смот­рим из­ме­не­ние b:

while x>0 do begin

b:=b\*(x mod 100);

end;

Опе­ра­тор mod воз­вра­ща­ет оста­ток от де­ле­ния, при де­ле­нии на 100 это по­след­ние две цифры *x*. Сле­до­ва­тель­но, по­след­няя цифра числа долж­на быть 7, а во вто­ром раз­ря­де дол­жен быть либо ноль либо еди­ни­ца, а в пер­вом — еди­ни­ца. Ис­ко­мое число — 107.

Ответ: 107.

**3.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет число *M*. Из­вест­но, что *x* > 100. Ука­жи­те наи­мень­шее такое (т.е. боль­шее 100) число *x*, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет 26.

|  |
| --- |
| var x, L, M: integer;  begin      readln(x);      L := x;      M := 65;      if L mod 2 = 0 then          M := 52;      while L <> M do          if L > M then              L := L - M          else              M := M – L;      writeln(M);  end. |

**По­яс­не­ние.**

В теле цикла числа M и L умень­ша­ют­ся, пока не ста­нут рав­ны­ми. Чтобы в итоге было на­пе­ча­та­но 26, оба числа в какой-то мо­мент долж­ны быть равны 26. Пой­дем от конца к на­ча­лу: на преды­ду­щем шаге одно число было 26, а дру­гое 26 + 26 = 52. Еще на шаг рань­ше 52 + 26 = 78 и 52. До того 78 + 52 = 130 и 52. То есть наи­мень­шее воз­мож­ное число 130. А по­сколь­ку най­ден­ное число чет­ное, то M будет при­сво­е­но зна­че­ние 52, что и при­ве­дет к не­об­хо­ди­мо­му ре­зуль­та­ту.

Ответ: 130.

**4.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 48, а потом 6.

|  |
| --- |
| var x, a, b, c: integer;  begin    readln(x);    a := 1; b := 0;    while x>0 do    begin      c := x mod 10;      a := a\*c;      if c>b then b := c;      x := x div 10;    end;    writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной *x*:

while x > 0 do begin

...

x:= x div 10;

end;

 Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся в *c* раз. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 48, не­об­хо­ди­мо, чтобы про­из­ве­де­ние цифр числа *x* была равна 48. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 6, не­об­хо­ди­мо, чтобы боль­шей циф­рой числа *x* была цифра 6. Наи­мень­шее такое число — 246.

Ответ: 246.

**5.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 45, а потом 5.

|  |
| --- |
| var x, a, b, c: integer;  begin    readln(x);    a := 1; b := 10;    while x>0 do    begin      c := x mod 10;    a := a\*c;      if c < b then b := c;      x := x div 10;    end;    writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной *x*:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся в *c* раз. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 45, не­об­хо­ди­мо, чтобы про­из­ве­де­ние цифр числа *x* была равна 45. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 5, не­об­хо­ди­мо, чтобы мень­шей циф­рой числа *x* была цифра 5. Наи­мень­шее такое число — 59. Ответ: 59.

**6.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет число *M*. Из­вест­но, что *x* > 100. Ука­жи­те наи­мень­шее такое (т. е. боль­шее 100) число *x*, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет 30.

|  |
| --- |
| var x, L, M: integer;  begin    readln(x);    L := x-30;    M := x+30;    while L <> M do      if L > M then        L := L - M      else        M := M – L;    writeln(M);  end. |

**По­яс­не­ние.**

В теле цикла числа M и L умень­ша­ют­ся, пока не ста­нут рав­ны­ми. Чтобы в итоге было на­пе­ча­та­но 30, оба числа в какой-то мо­мент долж­ны быть равны 30. Пой­дем от конца к на­ча­лу: на преды­ду­щем шаге одно число было 30, а дру­гое 30 + 30 = 60. Еще на шаг рань­ше 60 + 30 = 90 и 30. До того 90 + 30 = 120 и 90. То есть наи­мень­шее воз­мож­ное число 120. Ответ: 120.

**7.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет число *M*. Из­вест­но, что *x* > 100. Ука­жи­те наи­мень­шее такое (т. е. боль­шее 100) число *x*, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет 15.

|  |
| --- |
| var x, L, M: integer;  begin    readln(x);    L := x-30;    M := x+30;    while L <> M do      if L > M then        L := L - M      else        M := M – L; writeln(M); end. |

**По­яс­не­ние.**

В теле цикла числа M и L умень­ша­ют­ся, пока не ста­нут рав­ны­ми. Чтобы в итоге было на­пе­ча­та­но 15, оба числа в какой-то мо­мент долж­ны быть равны 15, а раз­ность пер­вых двух долж­на быть равна 60. Пой­дем от конца к на­ча­лу: на преды­ду­щем шаге одно число было 15, а дру­гое 15 + 15 = 30. Еще на шаг рань­ше 30 + 15 = 45 и 30. До того 45 + 15 = 60 и 15. Далее 60 + 15 = 75 и 60. Еще ранее 75 + 60 = 135 и 75. 135 - 75 = 60. То есть наи­мень­шее воз­мож­ное число 135 - 30 = 105.

Ответ: 105.

**8.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет числа: a и b. Ука­жи­те наи­мень­шее по­ло­жи­тель­ное пя­ти­знач­ное число x, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет сна­ча­ла 6, а потом 3.

|  |
| --- |
|  |
| var x, y, a, b: integer;  begin      a := 0;      b := 10;      readln(x);      while x > 0 do          begin              y := x mod 10;              x := x div 10              if y > a then                      a := y;              if y < b then                      b := y;          end;  writeln(a);  writeln(b)  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

y := x mod 10;

x:= x div 10;

...

end;

Т. к. опе­ра­тор div воз­вра­ща­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры. Сле­до­ва­тель­но, цикл вы­пол­нится пять раз.

В пе­ре­мен­ную a за­пи­сы­ва­ет­ся оста­ток от де­ле­ния числа x на 10 при усло­вии, что этот оста­ток боль­ше числа уже за­пи­сан­но­го в пе­ре­мен­ной a. В пе­ре­мен­ную b за­пи­сы­ва­ет­ся оста­ток от де­ле­ния числа числа x на 10 при усло­вии, что этот оста­ток мень­ше числа уже за­пи­сан­но­го в пе­ре­мен­ной b. Таким об­ра­зом, в пе­ре­мен­ную a за­пи­сы­ва­ет­ся наи­боль­шая цифра числа x, а в пе­ре­мен­ную b — наи­мень­шая. Зна­чит, наи­мень­шее число, удо­вле­тво­ря­ю­щее усло­вию задчи — 33336.

Ответ: 33336.

**9.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 0.

var x, L, M: integer;

begin

    readln(x);

    L:=0; M:=0;

    while x > 0 do begin

        L:= L + 1;

        if x mod 2 = 0 then

            M:= M + x mod 10;

        x:= x div 10;

    end;

    writeln(L); write(M);

end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число L столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние L сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы L стало L=3, *x* долж­но быть **трёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния M:

if x mod 2 = 0 then

  M:= M + x mod 10;

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*.

Чтобы зна­че­ние M оста­лось ну­ле­вым, нужно, чтобы усло­вие x mod 2 = 0 не вы­пол­ня­лось, либо чтобы вы­пол­ня­лось x mod 10 = 0. А зна­чит, все цифры этого числа могут быть либо нечётными, либо пер­вая цифра нечётная и остль­ные нули. Но мы ищем ми­ни­маль­ное *x*, по­это­му ис­ко­мое число 100.

Ответ: 100.

**10.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 8.

var x, L, M: integer;

begin

    readln(x);

    L:=0; M:=0;

    while x > 0 do begin

        L:=L+1;

        if (M < x) and (x mod 2 = 0) then begin

            M:=x mod 10;

        end;

    x:= x div 10;

    end;

    writeln(L); write(M);

end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число L столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние L сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы L стало L=3, *x* долж­но быть **трёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния M:

if (M < x) and (x mod 2 = 0) then begin

  M:=x mod 10;

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*.

Уже на пер­вом шаге можно по­лу­чить x mod 10 = 8, если по­след­няя цифра числа *x* есть 8. Далее можно сде­лать так, чтобы M боль­ше не ме­ня­лось: для этого на­ру­шим усло­вие x mod 2 = 0, т. е. сде­ла­ем пер­вые две цифры ис­ход­но­го числа *x* нечётными, при этом нам уже не будет важно усло­вие M < x. Тогда наи­мень­шее зна­че­ние вво­ди­мо­го числа *x* = 118.

Ответ: 118.

**11.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 72.

var x, a, b: integer;

begin

    readln(x);

    a:=0; b:=1;

    while x>0 do begin

        a:=a+1;

        b:=b\*(x mod 10);

        x:= x div 10;

    end;

    writeln(a); write(b);

end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы a стало a = 2, *x* долж­но быть **дву­знач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

b:=b\*(x mod 10);

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*; сле­до­ва­тель­но, число b по­лу­ча­ет­ся про­из­ве­де­ни­ем цифр числа *x*.

Пред­ста­вим число 72 в виде: 72 = 9 \* 8. Дру­го­го пред­став­ле­ния, в ко­то­ром оба мно­жи­те­ля од­но­знач­ные число 72 не имеет. Сле­до­ва­тель­но, ми­ни­маль­ное число *x* = 89.

Ответ: 89.

**12.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 7.

var x, L, M: integer;

begin

    readln(x);

    L:=0; M:=0;

    while x > 0 do begin

        L:= L + 1;

        if x mod 2 = 1 then

            M:= M + (x mod 10) div 2;

        x:= x div 10;

    end;

    writeln(L); write(M);

end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число L столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние L сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы L стало L=3, *x* долж­но быть **трёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния M:

if x mod 2 = 1 then

M:= M + (x mod 10) div 2;

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*.

 x mod 10 при­ни­ма­ет зна­че­ния от 0 до 9, сле­до­ва­тель­но, (x mod 10) div 2 — от 0 до 4. От­сю­да сле­ду­ет, что для по­лу­че­ния числа 7 нужно не менее двух сл­га­е­мых, по­лу­чен­ных в цикле как (x mod 10) div 2. Для этого не­об­хо­ди­мо, чтобы хотя бы две цифры в числе *x* были нечётными.

Пред­ста­вим 7 в виде: 7 = 4 + 3.

Для по­лу­че­ния наи­мень­ше­го *x* по­ло­жим первую цифру *x*(1) = 1, она даст вклад 0 на тре­тьем шаге цикла, а вто­рую цифру *x*(2) = 7, она даст вклад 3 на вто­ром шаге цикла. Остаётся сде­лать *x*(3) = 9, и по­лу­чим окон­ча­тель­ный ответ: *x* = 179. Ответ: 179.

**13.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 7.

var x, a, b : integer;

begin

    readln(x);

    a := 0; b := 1;

    while x > 0 do begin

        a := a + 1;

        b := b \* (x mod 10);

        x := x div 10;

    end;

    writeln(a); write(b); end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифр.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы a стало a = 3, *x* долж­но быть **трёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

b:=b\*(x mod 10);

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*; сле­до­ва­тель­но, число b по­лу­ча­ет­ся про­из­ве­де­ни­ем цифр числа *x*.

b = *x*(1) \* *x*(2) \* *x*(3) = 7, где число в скоб­ках озна­ча­ет номер цифры в числе *x*. Число 7 про­стое, по­это­му 7 = 7 \* 1 \* 1. Со­став­ля­ем ми­ни­маль­ное число: 117

Ответ: 117.

**14.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 28.

var x, L, M: integer;

begin

    readln(x);

    L:=0; M:=0;

    while x > 0 do begin

        L:=L+1;

        if M < x then begin

            M:= M + (x mod 10) \* 2;

        end;

        x:= x div 10;

    end;

    writeln(L); write(M);

end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число L столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние L сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы L стало L=3, *x* долж­но быть **трёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния M:

if M < x then begin

M:= M + (x mod 10) \* 2;

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*. Таким об­ра­зом М есть сумма удво­ен­ных цифр числа *x*, ко­то­рые при­ни­ма­ют чётные зна­че­ния от 0 до 18.

Для по­лу­че­ния наи­мень­ше­го *x* по­ло­жим первую цифру *x*(1) = 1, она не даст вклад на тре­тьем шаге цикла, по­то­му что не будет вы­пол­нять­ся усло­вие M < x, а для того, чтобы сумма удво­ен­ных цифр была равна 28, вто­рая и треть­ся цифры долж­ны быть боль­ше нуля.

Сде­ла­ем цифру *x*(3) мак­си­маль­но во­мож­ной, чтобы умень­шить вто­рую цифру и до­стичь ми­ни­му­ма:

*x*(3) = 9, тогда 9 \* 2 = 18, а *x*(2) = (28 - 18) / 2 = 5, но тогда не вы­пол­ня­ет­ся усло­вие M < x (18 > 15) и зна­че­ние 28 не до­стиг­нет­ся.

*x*(3) = 8, тогда 8 \* 2 = 16, а *x*(2) = (28 - 16) / 2 = 6, усло­вие M < x не вы­пол­ня­ет­ся (16 = 16) и зна­че­ние 28 не до­стиг­нет­ся.

*x*(3) = 7, тогда 7 \* 2 = 14, а *x*(2) = (28 - 14) / 2 = 7, M < x (14 < 17) и зна­че­ние 28 до­стигaется.

Окон­ча­тель­но на­хо­дим наи­мень­шее число *x*: 177.

Ответ: 177.

**15.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 8.

var x, L, M: integer;

begin

    readln(x);

    L:=0; M:=0;

    while x > 0 do begin

        L:= L + 1;

        if x mod 2 = 0 then

            M:= M + x mod 10;

        x:= x div 10;

    end;

    writeln(L); write(M);

end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число L столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние L сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы L стало L=3, *x* долж­но быть **трёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния M:

if x mod 2 = 0 then

  M:= M + x mod 10;

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*.

Усло­вие x mod 2 = 0 озна­ча­ет сле­ду­ю­щее: чтобы M уве­ли­чи­лось, число *x* долж­но со­дер­жать чётные цифры. Сле­до­ва­тель­но пе­ре­мен­ная М есть сумма чётных цифр числа *x*.

По­сколь­ку мы хотим по­лу­чить наи­мень­шее число, то по­ло­жим первую цифру в числе *x* рав­ной 1.

Оста­ток от де­ле­ния на 10 при­ни­ма­ет зна­че­ния от 0 до 9. Зна­чит мы можем по­лу­чить М = 8, уже на пер­вом шаге, если по­след­няя цифра равна 8. Чтобы не уве­ли­чить число М вто­рая цифра долж­на быть либо нечётной, либо нулём. Ис­хо­дя из усло­вия, по­лу­ча­ем ис­ко­мое число 108.

Ответ: 108.

**16.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 10.

var x, L, M: integer;

begin

    readln(x);

    L:=0; M:=0;

    while x > 0 do begin

        L:=L+1;

        if (M < x) and (x mod 2 = 1) then begin

            M:= (x mod 10) \* 2;

        end;

        x:= x div 10;

    end;

    writeln(L); write(M);

end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число L столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние L сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы L стало L=3, *x* долж­но быть **трёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния M:

if (M < x) and (x mod 2 = 1) then begin

  M:= (x mod 10) \* 2;

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*.

Чтобы по­лу­чить M = 10, в числе *x* долж­на быть цифра 5. Это можно сде­лать уже на пер­вом шаге. На вто­ром шаге число *x* уже дву­знач­ное. Если не будет вы­пол­нять­ся усло­вие M < x, то M далее не из­ме­нит­ся. Это воз­мож­но толь­ко при *x* = 105, по­сколь­ку 10 наи­мень­шее из дву­знач­ных чисел, при этом мы по­лу­чим наи­мень­шее ис­ко­мое число.

Ответ: 105.

**17.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 36.

var x, a, b : integer;

begin

    readln(x);

    a := 0; b := 1;

    while x > 0 do begin

        a := a + 1;

        b := b \* (x mod 10);

        x := x div 10;

    end;

    writeln(a); write(b);

end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы a стало a = 3, *x* долж­но быть **трёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

b:=b\*(x mod 10);

end;

Опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния, при де­ле­нии на 10, это по­след­няя цифра *x*; сле­до­ва­тель­но, число b по­лу­ча­ет­ся про­из­ве­де­ни­ем цифр числа *x*.

Пред­ста­вим число 36 в виде: 36 = 6 · 6 · 1 = 4 · 9 · 1 = 9 · 2 · 2 = 3 · 2 · 6. Дру­го­го пред­став­ле­ния, в ко­то­ром все мно­жи­те­ли од­но­знач­ные число 36 не имеет. Мы хотим, чтобы *x* было наи­мень­шим, по­это­му вы­бе­рем первую цифру рав­ной 1. Тогда остав­шя­ся пара цифр может быть 9 и 4, либо 6 и 6. Со­ста­вим из них наи­мень­шее число и при­пи­шем его к еди­ни­це. В ре­зуль­та­те по­лу­чим 149.

Ответ: 149.

**18.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 14.

var x, a, b : integer;

begin

    readln(x);

    a := 0; b := 1;

    while x > 0 do begin

        a := a + 1;

        b := b \* (x mod 10);

        x := x div 10;

    end;

    writeln(a); write(b);

end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы a стало a = 2, *x* долж­но быть **дву­знач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

b:=b\*(x mod 10);

end;

Опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*; сле­до­ва­тель­но, число b по­лу­ча­ет­ся про­из­ве­де­ни­ем цифр числа *x*.

Пред­ста­вим число 14 в виде: 14 = 7 \* 2. Дру­го­го пред­став­ле­ния, в ко­то­ром оба мно­жи­те­ля од­но­знач­ные число 14 не имеет. Сле­до­ва­тель­но, ми­ни­маль­ное число *x* = 27.

Ответ: 27.

**19.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 7.

var x, L, M: integer;

begin

    readln(x);

    L:=0; M:=0;

    while x > 0 do begin

        L:=L+1;

        if M < x then begin

            M:=x mod 10;

        end;

        x:= x div 10;

    end;

    writeln(L); write(M); end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число L столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние L сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы L стало L = 3, *x* долж­но быть **трёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния М:

if M < x then begin

M:=x mod 10;

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра.

После пер­во­го шага M может быть любым од­но­знач­ным чис­лом, причём на вто­ром шаге число *x* дву­знач­ное, а од­но­знач­ное число все­гда мень­ше лю­бо­го дву­знач­но­го, а зна­чит, M в любом слу­чае по­ме­ня­ет своё зна­че­ние. По­это­му вы­бе­рем тре­тью цифру числа *x* ми­ни­маль­ной *x*(3) = 0.

Вто­рая цифра уже не может быть любой, так как при *x*(2) > *x*(1) зна­че­ние пе­ре­мен­ной M уже не по­ме­ня­ет­ся на тре­тьем шаге. По­это­му *x*(2) не пре­вы­ша­ет 7. По­сколь­ку мы хотим по­лу­чить наи­мень­шее число *x*, мы берём *x*(1) = 1. Тогда *x*(2) обя­за­на быть рав­ной 7, в ре­зуль­та­те по­лу­а­ем число 170.

Ответ: 170.

**20.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 7.

var x, L, M: integer;

begin

    readln(x);

    L:=0; M:=0;

    while x > 0 do begin

        L:= L + 1;

        M:= M + x mod 10;

        x:= x div 10;

    end;

    writeln(L); write(M);

end.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число L столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние L сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы L стало L=3, *x* долж­но быть **трёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния M:

M:= M + x mod 10;

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*. Таким об­ра­зом М есть сумма цифр числа *x*.

  Для по­лу­че­ния наи­мень­ше­го *x* по­ло­жим первую цифру *x*(1) = 1, она даст вклад 1 на тре­тьем шаге цикла.

Пред­ста­вим 7 в виде суммы: 7 = 1 + 6 = 1 + 5 + 1 = 1 + 4 + 2 = 1 + 3 + 3. Если мы будем ис­поль­зо­вать сумму из двух сла­га­е­мых, то смо­жем за­ну­лить вто­рую цифру и тре­тья ока­жет­ся рав­ной 6. А т. к. 6 мень­ше лю­бо­го дву­знач­но­го числа, то и наи­мень­шее число *x* по­лу­чит­ся ран­вым 106.

Ответ: 106.

**21.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа a и b. Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 10.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a:=0; b:=0;      while x>0 do          begin              a:=a+1;              b:=b + (x mod 10);              x:=x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы a стало a = 2, *x* долж­но быть **дву­знач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

b:=b + (x mod 10);

end;

Опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*; сле­до­ва­тель­но, число b по­лу­ча­ет­ся сум­мой цифр числа *x*.

Чтобы по­лу­чить наи­мень­шее *x* по­ло­жим его первую цифру рав­ной *x*(1) = 1. Тогда вто­рая цифра *x*(2) будет равна 10 - 1 = 9, сле­до­ва­тель­но наи­мень­шее число *x* = 19.

Ответ: 19.

**22.** Ниже на 4-х язы­ках за­пи­сан ал­го­ритм. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа *a* и *b*.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a: = 0; b: = 1;      while x>0 do          begin              a : = a + 1;              b : = b \* (x mod 10);              x : = x div 10          end;      writeln(a);      write(b);  end. |

Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 14.

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си x от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в x. Для того, чтобы a стало a = 3, x долж­но быть **трёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

b:=b\*(x mod 10);

end;

Опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра x; сле­до­ва­тель­но, число b по­лу­ча­ет­ся про­из­ве­де­ни­ем цифр числа x.

Пред­ста­вим число 14 в виде: 14 = 7 \* 2 \* 1. Дру­го­го пред­став­ле­ния, в ко­то­ром все мно­жи­те­ли од­но­знач­ные число 14 не имеет. Сле­до­ва­тель­но, ми­ни­маль­ное число x = 127.

Ответ: 127.

**23.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа: a и b. Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 21.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a:=0; b:=1;      while x>0 do          begin              a:=a+1;              b:=b\*(x mod 10);              x:= x div 10          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си x от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в x. Для того, чтобы a стало a = 2, x долж­но быть дву­знач­ным.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

b:=b\*(x mod 10);

end;

Опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра x; сле­до­ва­тель­но, число b по­лу­ча­ет­ся про­из­ве­де­ни­ем цифр числа x.

Пред­ста­вим число 21 в виде: 21 = 7 \* 3. Дру­го­го пред­став­ле­ния, в ко­то­ром все мно­жи­те­ли од­но­знач­ные, число 21 не имеет. Сле­до­ва­тель­но, ми­ни­маль­ное число x = 37.

Ответ: 37.

**24.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число х, этот ал­го­ритм пе­ча­та­ет два числа а и Ь. Ука­жи­те наи­мень­шее из таких чисел х. при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3. а потом 7.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a: = 0; b: = 0;      while x>0 do          begin              a : = a + 1;              b : = b + (x mod 10);              x : = x div 10          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы a стало a = 3, *x* долж­но быть трех­знач­ным.

Те­перь рас­смот­рим из­ме­не­ние b:

while x>0 do begin

b:=b + (x mod 10);

end;

Опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*; сле­до­ва­тель­но, число b по­лу­ча­ет­ся сум­мой цифр числа *x*.

Чтобы по­лу­чить наи­мень­шее *x* по­ло­жим его пер­вые две цифры нулем и ше­стью. Тогда тре­тья цифра *x*(3) будет равна 7 − 2 = 5, сле­до­ва­тель­но, наи­мень­шее число *x* = 106.

Ответ: 106.

**25.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа a и b.   
Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 6, а потом 5.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              a := a + 2;              b := b + (x mod 10);              x:= x div 10;          end;      writeln(a); write(b); end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си x от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 2. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в x. Для того, чтобы a стало рав­ным 6, x долж­но быть трех­знач­ным.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

b:=b+(x mod 10);

end;

Опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра x; сле­до­ва­тель­но, число b по­лу­ча­ет­ся сум­мой цифр числа x.

Пред­ста­вим число 5 в виде: 5 = 1+0+4. Это не един­ствен­ные цифры, с по­мо­щью суммы ко­то­рых можно пред­ста­вить число 5, но они дают ми­ни­маль­ное трех­знач­ное число - 104. Сле­до­ва­тель­но, x = 104.

Ответ: 104.

**26.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа a и b.   
Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 6, а потом 9.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              a := a + 2;              b := b + (x mod 10);              x:= x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си x от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 2. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в x. Для того, чтобы a стало рав­ным 6, x долж­но быть трех­знач­ным.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

b:=b+(x mod 10);

end;

Опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра x; сле­до­ва­тель­но, число b по­лу­ча­ет­ся сум­мой цифр числа x.

Пред­ста­вим число 9 в виде: 9 = 1+0+8. Это не един­ствен­ные цифры, с по­мо­щью суммы ко­то­рых можно пред­ста­вить число , но они дают ми­ни­маль­ное трех­знач­ное число - 108. Сле­до­ва­тель­но, x = 108.

Ответ: 108.

**27.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число N, этот ал­го­ритм пе­ча­та­ет число q. Ука­жи­те наи­мень­шее из таких чисел N, при вводе ко­то­рых ал­го­ритм на­пе­ча­та­ет 17.

|  |
| --- |
| var N, q, i: integer;  begin      read(N);      for i : = 1 to N - 1 do begin          if N mod i = 0 then q : = i      end;      write(q)  end. |

**По­яс­не­ние.**

Фраг­мент кода на­хо­дит наи­боль­ший де­ли­тель N,не рав­ный са­мо­му числу N. Для наи­мень­ше­го N де­ли­тель равен 2, сле­до­ва­тель­но, N= 2 · 17 = 34.

**28.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число N, этот ал­го­ритм пе­ча­та­ет число q.   
Ука­жи­те наи­мень­шее из таких чисел N, при вводе ко­то­ро­го ал­го­ритм на­пе­ча­та­ет 13.

|  |
| --- |
| var N, q, i: integer;  begin  read(N);  for i : = 1 to N - 1 do begin  if N mod i = 0 then q : = i  end;  write(q)  end. |

**По­яс­не­ние.**

Фраг­мент кода на­хо­дит наи­боль­ший де­ли­тель N,не рав­ный са­мо­му числу N. Для наи­мень­ше­го N де­ли­тель равен 2, сле­до­ва­тель­но, N=13 · 2 = 26.

**29.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число https://ege.sdamgia.ru/formula/9d/9dd4e461268c8034f5c8564e155c67a6p.png, этот ал­го­ритм пе­ча­та­ет два числа: https://ege.sdamgia.ru/formula/0c/0cc175b9c0f1b6a831c399e269772661p.pngи https://ege.sdamgia.ru/formula/92/92eb5ffee6ae2fec3ad71c777531578fp.png. Ука­жи­те наи­мень­шее из таких чисел https://ege.sdamgia.ru/formula/9d/9dd4e461268c8034f5c8564e155c67a6p.png, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 3. До­пус­ка­ет­ся диа­па­зон зна­че­ний для ве­ли­чин це­ло­го типа: от −231 до 231 — 1.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              с := x mod 2;              if c = 0 then                  a := a + 1              else                  b := b + 1              x := x div 10          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, если число *x* крат­но двум, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на еди­ни­цу, в про­тив­ном слу­чае — пе­ре­мен­ная *b*. По­сколь­ку тре­бу­ет­ся, чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 2, а потом 3, не­об­хо­ди­мо, чтобы число *x* имело пять раз­ря­дов. Две цифры в этом числе долж­ны быть чётные, а три — нечётные. Ми­ни­маль­ным чис­лом, удо­вле­тво­ря­ю­щим дан­ным усло­ви­ям яв­ля­ет­ся число 10011.

Ответ: 10011.

**30.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 11, а потом 5.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 10;      while x>0 do          begin              с := x mod 10;              a := a+c;              if c < b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 11, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 11. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 5, не­об­хо­ди­мо, чтобы пер­вой циф­рой числа *x* была цифра 5. Наи­мень­шее такое число — 56.

Ответ: 56.

**31.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число https://ege.sdamgia.ru/formula/9d/9dd4e461268c8034f5c8564e155c67a6p.png, этот ал­го­ритм пе­ча­та­ет два числа: https://ege.sdamgia.ru/formula/0c/0cc175b9c0f1b6a831c399e269772661p.pngи https://ege.sdamgia.ru/formula/92/92eb5ffee6ae2fec3ad71c777531578fp.png. Ука­жи­те наи­мень­шее из таких чисел https://ege.sdamgia.ru/formula/9d/9dd4e461268c8034f5c8564e155c67a6p.png, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 2.

|  |
| --- |
| **Пас­каль** |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              с := x mod 2;              if c = 0 then                  a := a + 1              else                  b := b + 1              x := x div 10          end;      writeln(a); write(b);  end. |
|  |
|  |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, если число *x* крат­но двум, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на еди­ни­цу, в про­тив­ном слу­чае — пе­ре­мен­ная *b*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 3, а потом 2, не­об­хо­ди­мо, чтобы число *x* имело пять раз­ря­дов. Три цифры в этом числе долж­ны быть чётные, а две — нечётные. Ми­ни­маль­ным чис­лом, удо­вле­тво­ря­ю­щим дан­ным усло­ви­ям яв­ля­ет­ся число 10001.

Ответ: 10001.

**32.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число https://ege.sdamgia.ru/formula/9d/9dd4e461268c8034f5c8564e155c67a6p.png, этот ал­го­ритм пе­ча­та­ет два числа: https://ege.sdamgia.ru/formula/0c/0cc175b9c0f1b6a831c399e269772661p.pngи https://ege.sdamgia.ru/formula/92/92eb5ffee6ae2fec3ad71c777531578fp.png. Ука­жи­те наи­мень­шее из таких чисел https://ege.sdamgia.ru/formula/9d/9dd4e461268c8034f5c8564e155c67a6p.png, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 2. До­пус­ка­ет­ся диа­па­зон зна­че­ний для ве­ли­чин це­ло­го типа: от −231 до 231 — 1.

|  |
| --- |
|  |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              с := x mod 2;              if c = 0 then                  a := a + 1              else                  b := b + 1              x := x div 10          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

 while x > 0 do begin

...

x:= x div 10;

end;

 Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

 Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

 При этом, если число *x* крат­но двум, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на еди­ни­цу, в про­тив­ном слу­чае — пе­ре­мен­ная *b*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 3, а потом 2, не­об­хо­ди­мо, чтобы число *x* имело пять раз­ря­дов. Три цифры в этом числе долж­ны быть чётные, а две — нечётные. Ми­ни­маль­ным чис­лом, удо­вле­тво­ря­ю­щим дан­ным усло­ви­ям яв­ля­ет­ся число 10001.

 Ответ: 10001.

**33.** Ниже на четырёх язы­ках за­пи­сан ал­го­ритм. По­лу­чив на вход число https://ege.sdamgia.ru/formula/9d/9dd4e461268c8034f5c8564e155c67a6p.png, этот ал­го­ритм пе­ча­та­ет два числа: https://ege.sdamgia.ru/formula/0c/0cc175b9c0f1b6a831c399e269772661p.pngи https://ege.sdamgia.ru/formula/92/92eb5ffee6ae2fec3ad71c777531578fp.png. Ука­жи­те наи­мень­шее из таких чисел https://ege.sdamgia.ru/formula/9d/9dd4e461268c8034f5c8564e155c67a6p.png, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 3. До­пус­ка­ет­ся диа­па­зон зна­че­ний для ве­ли­чин це­ло­го типа: от −231 до 231 — 1.

|  |
| --- |
|  |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              с := x mod 2;              if c = 0 then                  a := a + 1              else                  b := b + 1              x := x div 10          end;      writeln(a); write(b); end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, если число *x* крат­но двум, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на еди­ни­цу, в про­тив­ном слу­чае — пе­ре­мен­ная *b*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 2, а потом 3, не­об­хо­ди­мо, чтобы число *x* имело пять раз­ря­дов. Две цифры в этом числе долж­ны быть чётные, а три — нечётные. Ми­ни­маль­ным чис­лом, удо­вле­тво­ря­ю­щим дан­ным усло­ви­ям яв­ля­ет­ся число 10011.

Ответ: 10011.

**34.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 11, а потом 5.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 10;      while x>0 do          begin              с := x mod 10;              a := a+c;              if c < b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

 while x > 0 do begin

...

x:= x div 10;

end;

 Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

 Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

 При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 11, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 11. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 5, не­об­хо­ди­мо, чтобы пер­вой циф­рой числа *x* была цифра 5. Наи­мень­шее такое число — 56.

 Ответ: 56.

**По­втор за­да­ния 5492.**

**35.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 12, а потом 5.

|  |
| --- |
|  |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 10;      while x>0 do          begin              с := x mod 10;              a := a+c;              if c < b then b := c;          x := x div 10;      end;      writeln(a); write(b);  end. |
|  |
|  |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

 while x > 0 do begin

...

x:= x div 10;

end;

 Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

 Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

 При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 12, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 12. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 5, не­об­хо­ди­мо, чтобы пер­вой циф­рой числа *x* была цифра 5. Наи­мень­шее такое число — 57.

 Ответ: 57.

**36.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 13, а потом 5.

|  |
| --- |
|  |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 10;      while x>0 do          begin              с := x mod 10;              a := a+c;              if c < b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |
|  |
|  |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

 while x > 0 do begin

...

x:= x div 10;

end;

 Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

 Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

 При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 13, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 13. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 5, не­об­хо­ди­мо, чтобы пер­вой циф­рой числа *x* была цифра 5. Наи­мень­шее такое число — 58.

 Ответ: 58.

**37.** Ниже на четырёх язы­ках за­пи­сан ал­го­ритм. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 14, а потом 5.

|  |
| --- |
|  |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 10;      while x>0 do          begin              с := x mod 10;              a := a+c;          if c < b then b := c;          x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

 while x > 0 do begin

...

x:= x div 10;

end;

 Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 14, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 14. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 5, не­об­хо­ди­мо, чтобы пер­вой циф­рой числа *x* была цифра 5. Наи­мень­шее такое число — 59.

Ответ: 59.

**38.** Ниже на четырёх язы­ках за­пи­сан ал­го­ритм. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 14, а потом 6.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 10;      while x>0 do          begin              с := x mod 10;              a := a+c;              if c < b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

 while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 14, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 14. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 6, не­об­хо­ди­мо, чтобы пер­вой циф­рой числа *x* была цифра 6. Наи­мень­шее такое число — 68.

Ответ: 68.

**39.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 16, а потом 7.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 10;      while x>0 do          begin              с := x mod 10;              a := a+c;              if c < b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

 while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 16, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 16. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 7, не­об­хо­ди­мо, чтобы пер­вой циф­рой числа *x* была цифра 7. Наи­мень­шее такое число — 79.

Ответ: 79.

**40.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 13, а потом 6.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 10;      while x>0 do          begin              с := x mod 10;              a := a+c;              if c < b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

 while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 13, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 13. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 6, не­об­хо­ди­мо, чтобы пер­вой циф­рой числа *x* была цифра 6. Наи­мень­шее такое число — 67.

 Ответ: 67.

**41.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 15, а потом 6.

|  |  |
| --- | --- |
|  | var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 10;      while x>0 do          begin              с := x mod 10;              a := a+c;              if c < b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 15, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 15. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 6, не­об­хо­ди­мо, чтобы пер­вой циф­рой числа *x* была цифра 6. Наи­мень­шее такое число — 69.

Ответ: 69.

**42.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 13, а потом 5.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 10;      while x>0 do          begin              c := x mod 10;              a := a+c;              if c < b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры. Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 мень­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 13, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 13. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 5, не­об­хо­ди­мо ,чтобы одной из циф­р числа *x* была цифра 5. Наи­мень­шее такое число — 58.

Ответ: 58.

**43.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 5.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a := 0; b := 0;          while x>0 do              begin                  a := a + 1;                  b := b + (x mod 10);                  x := x div 10;              end;          writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

По­сколь­ку опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си x от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом в пе­ре­мен­ная b уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния x на 10, а пе­ре­мен­ная a уве­ли­чи­ва­ет­ся на 1. По­сколь­ку тре­бу­ет­ся, чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 2, не­об­хо­ди­мо, чтобы число было дву­знач­ным. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 5, не­об­хо­ди­мо, чтобы сумма цифр числа x была равна 5. Наи­мень­шее такое число — 14.

Ответ: 14.

**44.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 9.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              a := a + 1;              b := b + (x mod 10);              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

По­сколь­ку опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си x от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом в пе­ре­мен­ная b уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния x на 10, а пе­ре­мен­ная a уве­ли­чи­ва­ет­ся на 1. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 2, не­об­хо­ди­мо, чтобы число было дву­знач­ным. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 9, не­об­хо­ди­мо, чтобы сумма цифр числа x была равна 9. Наи­мень­шее такое число — 18.

Ответ: 18.

**45.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 13, а потом 7.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              c := x mod 10;              a := a+c;              if c>b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры. Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 боль­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 13, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 13. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 7, не­об­хо­ди­мо, чтобы одной из циф­р числа *x* была цифра 7. Наи­мень­шее такое число — 67.

Ответ: 67.

**46.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 12, а потом 7.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              c := x mod 10;              a := a+c;              if c>b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры. Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 боль­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 12, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 12. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 7, не­об­хо­ди­мо, чтобы одной из циф­р числа *x* была цифра 7. Наи­мень­шее такое число — 57. Ответ: 57.

**47.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 11, а потом 7.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              c := x mod 10;              a := a+c;              if c>b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

 while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры. Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 боль­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 11, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 11. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 7, не­об­хо­ди­мо, чтобы одной из циф­р числа *x* была цифра 7. Наи­мень­шее такое число — 47.

Ответ: 47.

**48.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 10, а потом 7.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              c := x mod 10;              a := a+c;              if c>b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры. Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 боль­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 10, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 10. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 7, не­об­хо­ди­мо, чтобы одной из циф­р числа *x* была цифра 7. Наи­мень­шее такое число — 37.

Ответ: 37.

**49.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 9, а потом 7.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              c := x mod 10;              a := a+c;              if c>b then b := c;              x := x div 10;          end;          writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

 while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры. Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 боль­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 9, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 9. По­сколь­ку вто­рым долж­но быть на­пе­ча­та­но число 7, не­об­хо­ди­мо, чтобы одной из циф­р числа *x* была цифра 7. Наи­мень­шее такое число — 27.

Ответ: 27.

**50.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 9, а потом 5.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              c := x mod 10;              a := a+c;              if c>b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

По­сколь­ку опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры. Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 боль­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 9, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 9. Так как вто­рым долж­но быть на­пе­ча­та­но число 5, не­об­хо­ди­мо, чтобы одной из циф­р числа *x* была цифра 5. Наи­мень­шее такое число — 45.

Ответ: 45.

**51.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 8, а потом 5.

|  |
| --- |
| var x, a, b, c: integer;  begin      readln(x);      a := 0; b := 0;      while x>0 do          begin              c := x mod 10;              a := a+c;              if c>b then b := c;              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

 while x > 0 do begin

...

x:= x div 10;

end;

По­сколь­ку опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры. Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа.

При этом, пе­ре­мен­ной *c* при­сва­и­ва­ет­ся оста­ток от де­ле­ния *x* на 10, пе­ре­мен­ная *a* уве­ли­чи­ва­ет­ся на оста­ток от де­ле­ния х на 10. Если оста­ток от де­ле­ния *x* на 10 боль­ше пе­ре­мен­ной *b*, то пе­ре­мен­ной *b* при­сва­и­ва­ет­ся зна­че­ние пе­ре­мен­ной *c*. По­сколь­ку тре­бу­ет­ся ,чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла 8, не­об­хо­ди­мо, чтобы сумма цифр числа *x* была равна 8. Так как вто­рым долж­но быть на­пе­ча­та­но число 5, не­об­хо­ди­мо, чтобы одной из циф­р числа *x* была цифра 5. Наи­мень­шее такое число — 35.

Ответ: 35.

**52.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа: a и b. Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 2.

|  |
| --- |
| var x, a, b: integer;      begin      readln(x);      a:=0; b:=0;      while x>0 do          begin              a:=a + 1;              if b < (x mod 8)              then                  b:=x mod 8;              x:=x div 8;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Зна­че­ние в пе­ре­мен­ной *a* после вы­пол­не­ния цикла равно ко­ли­че­ству вы­пол­нен­ных цик­лов. По­сколь­ку тре­бу­ет­ся, чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла число 3, цикл дол­жен вы­пол­нить­ся три раза. Опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, сле­до­ва­тель­но, ис­ко­мое число долж­но два раза де­лить­ся на 8 так, чтобы оста­ток был не мень­ше вось­ми. Сле­до­ва­тель­но, это число долж­но быть не мень­ше числа 64.

В пе­ре­мен­ную *b* за­пи­сы­ва­ет­ся оста­ток от де­ле­ния числа на 8. По усло­вию тре­бу­ет­ся, чтобы после вы­пол­не­ния цикла пе­ре­мен­ная *b* имела зна­че­ние 2, т. е. оста­ток от по­след­не­го де­ле­ния на 8 в цикле дол­жен быть равен 2.

Вы­пол­ним про­грам­му для всех чисел, не мень­ших чем 64. Пер­вое число, ко­то­рое удо­вле­тво­рит усло­вию и будет наи­мень­шим. По­сколь­ку про­грам­ма вы­во­дит целые числа и ни­ка­ких дру­гих опе­ра­то­ров к числу, кроме опе­ра­то­ров div и mod, не при­ме­ня­ет­ся, будем рас­смат­ри­вать толь­ко целые числа.

При вводе числа 64 про­грам­ма вы­ве­дет числа 3 и 0. При вводе числа 65 про­грам­ма вы­ве­дет числа 3 и 1. При вводе числа 66 про­грам­ма вы­ве­дет числа 3 и 2. Сле­до­ва­тель­но, ответ 66.

Ответ: 66.

**53.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа: a и b. Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 4.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a:=0; b:=0;      while x>0 do          begin              a:=a + 1;              if b < (x mod 8)                  then b:=x mod 8;              x:=x div 8;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Зна­че­ние в пе­ре­мен­ной *a* после вы­пол­не­ния цикла равно ко­ли­че­ству вы­пол­нен­ных цик­лов. По­сколь­ку тре­бу­ет­ся, чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла число 3, цикл дол­жен вы­пол­нить­ся три раза. Опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, сле­до­ва­тель­но, ис­ко­мое число долж­но два раза де­лить­ся на 8 так, чтобы оста­ток был не мень­ше вось­ми. Сле­до­ва­тель­но, это число долж­но быть не мень­ше числа 64.

В пе­ре­мен­ную *b* за­пи­сы­ва­ет­ся оста­ток от де­ле­ния числа на 8. По усло­вию тре­бу­ет­ся, чтобы после вы­пол­не­ния цикла пе­ре­мен­ная *b* имела зна­че­ние 4, т. е. оста­ток от по­след­не­го де­ле­ния на 8 в цикле дол­жен быть равен 4.

Вы­пол­ним про­грам­му для всех чисел, не мень­ших чем 64. Пер­вое число, ко­то­рое удо­вле­тво­рит усло­вию и будет наи­мень­шим. По­сколь­ку про­грам­ма вы­во­дит целые числа и ни­ка­ких дру­гих опе­ра­то­ров к числу, кроме опе­ра­то­ров div и mod, не при­ме­ня­ет­ся, будем рас­смат­ри­вать толь­ко целые числа.

При вводе числа 64 про­грам­ма вы­ве­дет числа 3 и 0. При вводе числа 65 про­грам­ма вы­ве­дет числа 3 и 1. При вводе числа 66 про­грам­ма вы­ве­дет числа 3 и 2, 67 — 3 и 3, 68 — 3 и 4. Сле­до­ва­тель­но, ответ 68.

Ответ: 68.

**54.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа: a и b. Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 13.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a:=0; b:=0;      while x>0 do          begin              a:=a + 1;              b:=b + (x mod 10);              x:=x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Зна­че­ние в пе­ре­мен­ной *a* после вы­пол­не­ния цикла равно ко­ли­че­ству вы­пол­нен­ных цик­лов. По­сколь­ку тре­бу­ет­ся, чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла число 3, цикл дол­жен вы­пол­нить­ся три раза. Опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, сле­до­ва­тель­но, ис­ко­мое число долж­но два раза де­лить­ся на 10 так, чтобы оста­ток был не мень­ше де­ся­ти, то есть быть трёхзнач­ным.

В пе­ре­мен­ной *b* на­кап­ли­ва­ет­ся сумма цифр ис­ко­мо­го трёхзнач­но­го числа. Таким об­ра­зом, не­об­хо­ди­мо найти наи­мень­шее трёхзнач­ное число, сумма цифр ко­то­ро­го равна 13. Это число 139.

Ответ: 139.

**55.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа: a и b. Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 12.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a:=0; b:=0;      while x>0 do          begin              a:=a + 1;              b:=b + (x mod 10);              x:=x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Зна­че­ние в пе­ре­мен­ной *a* после вы­пол­не­ния цикла равно ко­ли­че­ству вы­пол­нен­ных цик­лов. По­сколь­ку тре­бу­ет­ся, чтобы про­грам­ма на­пе­ча­та­ла сна­ча­ла число 3, цикл дол­жен вы­пол­нить­ся три раза. Опе­ра­тор div остав­ля­ет толь­ко целую часть от де­ле­ния, сле­до­ва­тель­но, ис­ко­мое число долж­но два раза де­лить­ся на 10 так, чтобы оста­ток был не мень­ше де­ся­ти, то есть быть трёхзнач­ным.

В пе­ре­мен­ной *b* на­кап­ли­ва­ет­ся сумма цифр ис­ко­мо­го трёхзнач­но­го числа. Таким об­ра­зом, не­об­хо­ди­мо найти наи­мень­шее трёхзнач­ное число, сумма цифр ко­то­ро­го равна 12. Это число 129.

Ответ: 129.

**56.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа a и b.   
Ука­жи­те наи­мень­шее из таких чисел x, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет

сна­ча­ла 3, а потом 25.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a := 0; b := 0;      while x > 0 do          begin              a := a + 1;              if(x mod 2) <> 0 then                  b := b+(x mod 10);              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы a стало a = 3, *x* долж­но быть **трёхзнач­ным**.

 Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

...

if(x mod 2) <> 0 then

b := b+(x mod 10);

end;

Опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*. Усло­вие x mod 2 <> 0, озна­ча­ет, что *x* не де­лит­ся на­це­ло на 2, то есть яв­ля­ет­ся нечётным. Сле­до­ва­тель­но число b уве­ли­чи­ва­ет­ся на ве­ли­чи­ну по­след­ней цифры числа *x* толь­ко в том слу­чае, если оно нечётно. Наи­мень­шее *x*, име­ю­щее сумму цифр 25 и удо­вле­тво­ря­ю­щее усло­ви­ям за­да­чи равно 799.

Ответ: 799.

**57.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число , эта про­грам­ма пе­ча­та­ет два числа, L и M. Ука­жи­те наи­мень­шее из таких чисел , при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 8.

var x, L, M: integer;

begin

readln(x);

L:=0; M:=0;

while x > 0 do begin

L:= L + 1;

if x mod 2 = 1 then

M:= M + x mod 10;

x:= x div 10;

end;

writeln(L); write(M);

end.

**По­яс­не­ние.**

Строч­ка L:= L + 1; в дан­ной про­грам­ме счи­та­ет ко­ли­че­ство цифр в числе, зна­чит число трех­знач­ное.

При по­мо­щи ко­манд if x mod 2 = 1 then M:= M + x mod 10; под­счи­ты­ва­ет­ся сумма всех цифр числа x, если оно нечётно. Наи­мень­шее число, удо­вле­тво­ря­ю­щее усло­вию — 107.

Ответ: 107.

**58.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа *a* и *b*. Ука­жи­те такое наи­мень­шее число *x*, что при вводе x ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 0.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a := 0; b := 1;      while x > 0 do          begin              a := a+1;              b := b\*(x mod 10);              x := x div 10;          end;      writeln(a); write(b);  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

Из при­ве­ден­но­го цикла видно, что на каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы a стало равно 2, *x* долж­но быть **дву­знач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

...

b := b\*(x mod 10);

end;

Опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*. Сле­до­ва­тель­но число *b* умно­жа­ет­ся на ве­ли­чи­ну по­след­ней цифры числа *x*. Наи­мень­шее *x* удо­вле­тво­ря­ю­щее усло­ви­ям за­да­чи равно 10. Ответ: 10.

**59.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 15.

|  |
| --- |
| var x, a, b: integer;  begin      readln(x);      a:=0; b:=1;      while x>0 do          begin              a:=a+1;              b:=b\*(x mod 10);              x:= x div 10          end;      writeln(a); write(b)  end. |

**По­яс­не­ние.**

Рас­смот­рим цикл, число шагов ко­то­ро­го за­ви­сит от из­ме­не­ния пе­ре­мен­ной x:

while x > 0 do begin

...

x:= x div 10;

end;

Т. к. опе­ра­тор div воз­вра­ща­ет целую часть от де­ле­ния, то при де­ле­нии на 10 это рав­но­силь­но от­се­че­нию по­след­ней цифры.

На каж­дом шаге от де­ся­тич­ной за­пи­си *x* от­се­ка­ет­ся по­след­няя цифра до тех пор, пока все цифры не будут от­се­че­ны, то есть x не ста­нет равно 0. По­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом цифр в *x*. Для того, чтобы стало a = 2, *x* долж­но быть дву­знач­ным.

Те­перь рас­смот­рим из­ме­не­ние b:

while x>0 do begin

b:=b\*(x mod 10);

end;

Опе­ра­тор mod воз­вра­ща­ет оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*; сле­до­ва­тель­но, число b равно про­из­ве­де­нию цифр числа *x*. Пред­ста­вим число 15 в виде 5\*3. Дру­го­го пред­став­ле­ния, в ко­то­ром оба мно­жи­те­ля од­но­знач­ные, число 15 не имеет. Сле­до­ва­тель­но, ми­ни­маль­ное число *x* = 35.

Ответ: 35.

**60.** Ниже при­ве­де­ны тек­сты одной и той же про­грам­мы, за­пи­сан­ные на че­ты­рех язы­ках про­грам­ми­ро­ва­ния. Что будет на­пе­ча­та­но в ре­зуль­та­те вы­пол­не­ния этой про­грам­мы?

|  |
| --- |
| **Пас­каль** |
| Program Task;  Uses crt;  const L = 4;  type atype = array [1..L] of integer;  var R : atype;  N, p : integer;  Procedure Pr1(L : integer; var R : atype );      var i,n,t : integer;      begin          for i:=1 to L do              begin                  t:=(R[i] div 2)\*4;                  R[i]:=t mod 5;              end;      end;    Function F1 (L : integer; R: atype) : integer;      Var N, i, T : integer;      begin          N:=1;          T:=1;          for i:=1 to L do          begin              N:=N\*R[i]+T;              T:=T+2;          end;      F1:=N;  end;    BEGIN      R[1]:=6; R[2]:=10; R[3]:=7; R[4]:=3;      Pr1(L, R);      N:=F1(L,R);      write(N);      writeln;  END. |

*За­да­ние по­вы­шен­ной слож­но­сти.*

**По­яс­не­ние.**

В конце про­грам­мы на экран вы­во­дит­ся зна­че­ние пе­ре­мен­ной N, рас­смот­рим как оно по­лу­ча­ет­ся на при­ме­ре про­грам­мы на Пас­ка­ле.

По­сколь­ку L = 4, то из­на­чаль­но мы имеем од­но­мер­ный мас­сив из четырёх це­ло­чис­лен­ных элемн­тов R[i]:

R[1]:=6; R[2]:=10; R[3]:=7; R[4]:=3;. В ходе про­це­ду­ры Pr1 эти зна­че­ния из­ме­ня­ют­ся, причём не­за­ви­си­мо друг от друга:

for i:=1 to L do

begin

t:=(R[i] div 2)\*4;

R[i]:=t mod 5;

Опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, а опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния (при де­ле­нии на 5 он не пре­вы­ша­ет 4).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| R[i] | 6 | 10 | 7 | 3 |
| t = (R[i] div 2) \* 4 | 3 \* 4 | 5 \* 4 | 3 \* 4 | 1 \* 4 |
| R[i] = t mod 5 | 2 | 0 | 2 | 4 |

Функ­ции F1 при­сва­и­ва­ет­ся зна­че­ни пе­ре­ме­ной N, ко­то­рое внут­ри тела функ­ции по­лу­ча­ет­ся сле­ду­ю­щим об­ра­зом:

for i:=1 to L do

begin

N:=N\*R[i]+T;

T:=T+2;

end;

Под­ста­вим по­лу­чен­ные ранее зна­че­ния R[i]:

i=1: T = 1, N = 1 \* 2 + 1 = 3,

i=2: T = 3, N = 3 \* 0 + 3 = 3,

i=3: T = 5, N = 3 \* 2 + 5 = 11,

i=4: T = 7, N = 11 \* 4 + 7 = 51.

По­след­нее дей­ствие и даст нам ис­ко­мый ре­зуль­тат. Ответ: 51.

**61.** Ниже за­пи­са­на про­грам­ма. Что будет на­пе­ча­та­но в ре­зуль­та­те вы­пол­не­ния этой про­грам­мы

|  |
| --- |
| Program Task;  Uses crt;  const L = 4;  type atype = array [1..L] of integer;  Var R : atype;  N, p : integer;  Procedure Pr1(L : integer; var R : atype );      var i,n,t : integer;      begin          for i:=1 to L do          begin              t:=(R[i] div 2)\*4;              R[i]:=t mod 5;          end;      end;   Function F1 (L : integer; R: atype) : integer;      Var N, i, T : integer;      begin          N:=1;          T:=1;          for i:=1 to L do          begin              N:=N\*R[i]+T;              T:=T+2;          end;          F1:=N;      end;    BEGIN      R[1]:=5; R[2]:=9; R[3]:=6; R[4]:=2;      Pr1(L, R);      N:=F1(L,R);      write(N);     writeln; END. |

**По­яс­не­ние.**

В конце про­грам­мы на экран вы­во­дит­ся зна­че­ние пе­ре­мен­ной N, рас­смот­рим как она олу­ча­ет­ся на при­ме­ре про­грам­мы на Пас­ка­ле.

По­сколь­ку L = 4, то из­на­чаль­но мы имеем од­но­мер­ный мас­сив из четырёх це­ло­чис­лен­ных элемн­тов R[i]:

R[1]:=5; R[2]:=9; R[3]:=6; R[4]:=2;. В ходе про­це­ду­ры Pr1 эти зна­че­ния из­ме­ня­ют­ся, причём не­за­ви­си­мо друг от друга:

for i:=1 to L do

begin

t:=(R[i] div 2)\*4;

R[i]:=t mod 5;

Опе­ра­тор div остав­ля­ет целую часть от де­ле­ния, а опе­ра­тор mod остав­ля­ет оста­ток от де­ле­ния (при де­ле­нии на 5 он не пре­вы­ша­ет 4).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| R[i] | 5 | 9 | 6 | 2 |
| t = (R[i] div 2) \* 4 | 2 \* 4 | 4 \* 4 | 3 \* 4 | 1 \* 4 |
| R[i] = t mod 5 | 3 | 1 | 2 | 4 |

Функ­ции F1 при­сва­и­ва­ет­ся зна­че­ни пе­ре­ме­ной N, ко­то­рое внут­ри тела функ­ции по­лу­ча­ет­ся сле­ду­ю­щим об­ра­зом:

for i:=1 to L do

begin

N:=N\*R[i]+T;

T:=T+2;

end;

Под­ста­вим по­лу­чен­ные ранее зна­че­ния R[i]:

i=1: T = 1, N = 1 \* 3 + 1 = 4,

i=2: T = 3, N = 4 \* 1 + 3 = 7,

i=3: T = 5, N = 7 \* 2 + 5 = 19,

i=4: T = 7, N = 19 \* 4 + 7 = 83.

По­след­нее дей­ствие и даст нам ис­ко­мый ре­зуль­тат.

Ответ: 83.

**62.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 20.

|  |
| --- |
| program B20;  var x, a, b: integer;  begin    readln(x);    a := 0; b := 0;    while x > 0 do    begin      a := a + 1;      if x mod 2 = 0 then        b := b+ x mod 10      x := x div 10;    end;    writeln(a); write(b);  end. |

**По­яс­не­ние.**

Число а рав­ня­ет­ся ко­ли­че­ству цифр в х, а b - сумме чётных цифр х.

Есть всего два спо­со­ба пред­ста­вить 20 как сумму трёх чётных цифр: 20 = 4 + 8 + 8 = 6 + 6 + 8.

В пер­вом слу­чае число по­лу­чит­ся мень­ше, а имен­но 488.

**63.** Ниже на пяти язы­ках за­пи­сан ал­го­ритм. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа *a* и *b*. Ука­жи­те наи­мень­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 18.

|  |
| --- |
| program B20;  var x, a, b: integer;  begin    readln(x);    a := 0; b := 0;    while x > 0 do    begin      a := a + 1;      if x mod 2 = 0 then        b := b+ x mod 10;      x := x div 10;    end;    writeln(a); write(b);  end. |

**По­яс­не­ние.**

Число а рав­ня­ет­ся ко­ли­че­ству цифр в х, а b - сумме чётных цифр х.

Есть всего три спо­со­ба пред­ста­вить 18 как сумму трёх чётных цифр: 18 = 2 + 8 + 8 = 4 + 6 + 8 = 6 + 6 + 6.

В пер­вом слу­чае число по­лу­чит­ся мень­ше, а имен­но 288.

**64.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа *a* и *b*. Ука­жи­те наи­боль­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 10.

|  |
| --- |
| program B20;  var x, a, b: integer;  begin    readln(x);    a := 0; b := 0;    while x > 0 do    begin      a := a + 1;      if x mod 2 = 0 then        b := b + (x mod 10);      x := x div 10;    end;    writeln(a); write(b);  end. |

**По­яс­не­ние.**

Число а рав­ня­ет­ся ко­ли­че­ству цифр в х, а b - сумме чётных цифр х.

Чтобы число было по­боль­ше, в стар­ший раз­ряд по­ста­вим 9. Сумма остав­ших­ся двух цифр - 10, и они чётные.

10 = 2 + 8 = 4 + 6

82 > 64

В итоге по­лу­чи­ли число 982.

**65.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа *a* и *b*. Ука­жи­те наи­боль­шее из таких чисел *x*, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 12.

|  |
| --- |
| program B20;  var x, a, b: integer;  begin    readln(x);    a := 0; b := 0;    while x > 0 do    begin      a := a + 1;      if x mod 2 = 0 then        b := b + (x mod 10);      x := x div 10;    end;    writeln(a); write(b);  end. |

**По­яс­не­ние.**

Число а рав­ня­ет­ся ко­ли­че­ству цифр в х, а b - сумме чётных цифр х.

Чтобы число было по­боль­ше, в стар­ший раз­ряд по­ста­вим 9. Сумма остав­ших­ся двух цифр - 12, и они чётные.

12 = 4 + 8 = 6 + 6

84 > 66

В итоге по­лу­чи­ли число 984.

**66.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *х*, этот ал­го­ритм пе­ча­та­ет число М. Из­вест­но, что *х* > 40. Ука­жи­те наи­мень­шее такое (т. е. боль­шее 40) число х, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет 5.

|  |
| --- |
| var x, L, M: integer;  begin      readln(x);      L := x;      M := 5;      if L mod 2 = 0 then        M := 24;      while L <> M do        if L > M then          L := L − M        else          M = M − L      writeln(M);  end. |

**По­яс­не­ние.**

Цикл while — ничто иное, как ал­го­ритм Ев­кли­да, то есть он ищет наи­боль­ший общий де­ли­тель чисел M и L. В конце ра­бо­ты цикла M = L = НОД(M, L).

Если x — чётное число, то L = x, M = 24. Если нечётное, то L = x, M = 5.

24 не де­лит­ся на 5, по­это­му чтобы НОД(M, L) рав­нял­ся 5, нужно, M = 5, то есть чтобы x был нечётным.

Таким об­ра­зом, нам нужно самое ма­лень­кое нечётное число, ко­то­рое боль­ше 40 и де­лит­ся на 5. Это 45.

**67.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *х*, этот ал­го­ритм пе­ча­та­ет число М. Из­вест­но, что *х* > 40. Ука­жи­те наи­мень­шее такое (т. е. боль­шее 40) число х, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет 2.

|  |
| --- |
| var x, L, M: integer;  begin      readln(x);      L := x;      M := 12;      if L mod 2 = 0 then        M := 24;      while L <> M do        if L > M then          L := L − M        else          M = M − L      writeln(M);  end. |

**По­яс­не­ние.**

Цикл while — ничто иное, как ал­го­ритм Ев­кли­да, то есть он ищет наи­боль­ший общий де­ли­тель чисел M и L. В конце ра­бо­ты цикла M = L = НОД(M, L).

Если x — чётное число, то L = x, M = 24. Если нечётное, то L = x, M = 12.

При этом, если x — нечётное число, то НОД(L, M) https://ege.sdamgia.ru/formula/f9/f9bb70af966a4abbd08b776e6c5971adp.png2. По­это­му x дол­жен быть чётным.

Таким об­ра­зом, нам нужно самое ма­лень­кое чётное число, ко­то­рое боль­ше 40 и не де­лит­ся на 3 и 4 (тогда бы НОД(L, M) рав­нял­ся 6 или 4). Это число 46.

**68.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет число M. Из­вест­но, что x > 200. Ука­жи­те наи­мень­шее такое (т. е. боль­шее 200) число x, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет 60.

|  |
| --- |
| var x, L, M: integer;  begin    readln(x);    L := x-30;    M := x+30;    while L <> M do begin      if L > M        then L := L - M        else M := M – L;    end;    writeln(M);  end. |

**По­яс­не­ние.**

Цикл while — ничто иное, как ал­го­ритм Ев­кли­да, то есть он ищет наи­боль­ший общий де­ли­тель чисел M и L. В конце ра­бо­ты цикла M = L = НОД(M, L).

Зна­чит, нужно найти такое наи­мень­шее x боль­шее 200, чтобы числа L = x − 30 и M = x + 30 имели НОД рав­ный 60. При x = 210 L = 180 и M = 240, наи­боль­ший общий де­ли­тель равен 60. При 200 < x < 210, наи­боль­ший общий де­ли­тель L и M не может быть равен 60.

Ответ: 210.

**69.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход на­ту­раль­ное число *x*, этот ал­го­ритм пе­ча­та­ет число *R*. Ука­жи­те такое число *x*, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет дву­знач­ное число, сумма цифр ко­то­ро­го равна 16. Если таких чисел *x* не­сколь­ко, ука­жи­те наи­мень­шее из них.

|  |
| --- |
| var      x,d,R: longint;  begin      readln(x);      R := 0;      while x>0 do          begin              d := x mod 10;              R := 10\*R + d;              x := x div 10          end;      writeln(R)  end. |

**По­яс­не­ние.**

Дан­ный ал­го­ритм пе­ча­та­ет на вы­хо­де число x за­пи­сан­ное в об­рат­ном по­ряд­ке: то есть пер­вая цифра числа x на­хо­дит­ся на по­след­нем месте, вто­рая на пред­по­след­нем и так далее.

Раз­ло­жим 16 на мно­жи­те­ли так, чтобы одно из сла­га­е­мых было как можно боль­ше, а вто­рое как можно мень­ше: 16 = 7 + 9. Сле­до­ва­тель­но, наи­мень­шее число, удо­вле­тво­ря­ю­щее усло­вию за­да­чи — 79.

Ответ: 79.