**Алгоритмы, печатающие числа, наибольшее значение вводимого числа**

**1.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­боль­шее 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 = 0 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 = 0 then

  M:= M + (x mod 10) div 2;

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 быть чётным.

Пред­по­ло­жим, ис­ход­ное *x* нечётное, тогда на пер­вом шаге M = 0.

Если на вто­ром шаге *x* также нечётное (вто­рая цифра ис­ход­но­го числа нечётная), то M = 0, причём каким бы ни было зна­че­ние x на тре­тьем шаге, мы не смо­жем по­лу­чить M = 7, по­сколь­ку оста­ток от де­ле­ния чётного числа на 10 не пре­вос­хо­дит 8, а 8 / 2 = 4, сле­до­ва­тель­но, **вто­рая цифра** ис­ход­но­го *x* **чётная**.

Тогда пер­вая цифра может при­ни­мать зна­че­ния 2, 4, 6, 8, но мы ищем наи­боль­шее *x*, по­это­му сде­ла­ем первую цифру, рав­ной 9, тогда наше пред­по­ло­же­ние не удо­вле­тво­ря­ет усло­вию за­да­чи, и по­след­няя цифра ис­ход­но­го числа обя­за­на быть чётной, т.е. **ис­ход­ное *x* чётно**.

7 = 4 + 3, чему со­от­вет­ству­ют цифры 8 и 6. Те­перь, рас­по­ла­гая цифры по убы­ва­нию, на­хо­дим наи­боль­шее воз­мож­ное *x*: *x* = 986.

Ответ: 986.

 **2.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет числа: a и b. Ука­жи­те наи­боль­шее че­ты­рех­знач­ное число x, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет сна­ча­ла 5, а потом 7.

|  |
| --- |
| var x, y, a, b: integer;begin    a := 10;    b := 0;    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 че­ты­ре раза.

В пе­ре­мен­ную a за­пи­сы­ва­ет­ся оста­ток от де­ле­ния числа x на 10 при усло­вии, что этот оста­ток мень­ше числа уже за­пи­сан­но­го в пе­ре­мен­ной a. В пе­ре­мен­ную b за­пи­сы­ва­ет­ся оста­ток от де­ле­ния числа x на 10 при усло­вии, что этот оста­ток боль­ше числа уже за­пи­сан­но­го в пе­ре­мен­ной b. Таким об­ра­зом, в пе­ре­мен­ную a за­пи­сы­ва­ет­ся наи­боль­шая цифра числа x, а в пе­ре­мен­ную b — наи­мень­шая. Зна­чит, наи­боль­шее число, удо­вле­тво­ря­ю­щее усло­вию задчи — 7775. Ответ: 7775.

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

|  |
| --- |
| 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*. По­сколь­ку не­об­ход­мо, чтобы число было наи­боль­шим, пер­вая цифра числа долж­на быть5, во вто­ром раз­ря­де дол­жен быть либо ноль либо еди­ни­ца, а в пер­вом — еди­ни­ца. Ис­ко­мое число — 501.

Ответ: 501.

**4.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x*, этот ал­го­ритм пе­ча­та­ет два числа: *L* и *M*.
Ука­жи­те наи­мень­шее число *x*, при вводе ко­то­ро­го ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 6.

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

**По­яс­не­ние.**

Можно за­ме­тить, что число M = целая\_часть(log2x) + 1, а L — ко­ли­че­ство еди­ниц в дво­ич­ной за­пи­си числа x. Найдём ми­ни­маль­ное число, удо­вле­тво­ря­ю­щее этим усло­ви­ям. Это 1000112 = 35.

**5.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет числа: a и b.

Ука­жи­те наи­мень­шее по­ло­жи­тель­ное пя­ти­знач­ное число x, при ко­то­ром после вы­пол­не­ния ал­го­рит­ма будет на­пе­ча­та­но сна­ча­ла 4, а потом 2.

|  |
| --- |
| var x, y, a, b: integer;begin    a := 0;    b := 0;    readln(x);    while x > 0 do    begin        y := x mod 10;        if y > 3 then            a := a + 1;        if y < 8 then            b := b + 1;        x := x div 10    end;    writeln(a);    writeln(b)end. |

**По­яс­не­ние.**

Число — ко­ли­че­ство цифр в числе, боль­ших 3.

Число — ко­ли­че­ство цифр в числе, мень­ших 8.

Таким об­ра­зом, по­лу­ча­ет­ся 1 цифра мень­ше 4, цифра от 4 до 7 и 3 цифры боль­ше 7.

Под­би­рая для каж­дой из цифр ми­ни­маль­ное зна­че­ние и от­сор­ти­ро­вав эти цифры в по­ряд­ке воз­рас­та­ния, по­лу­чим ми­ни­маль­ное число 14888.

**6.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет числа: *a* и *b*.

Ука­жи­те наи­боль­шее по­ло­жи­тель­ное пя­ти­знач­ное число x, при ко­то­ром после вы­пол­не­ния ал­го­рит­ма будет на­пе­ча­та­но сна­ча­ла 3, а потом 4.

|  |
| --- |
| var x, y, a, b: integer;begin    a := 0;    b := 0;    readln(x);    while x > 0 do    begin        y := x mod 10;        if y > 4 then            a := a + 1;        if y < 7 then            b := b + 1;        x := x div 10    end;    writeln(a);    writeln(b)end. |

**По­яс­не­ние.**

а − ко­ли­че­ство цифр в числе, боль­ших, чем 4. b − ко­ли­че­ство цифр в числе, мень­ших, чем 7.

В нашем числе по­лу­ча­ет­ся одна цифра не мень­ше 7, две цифры не боль­ше, чем 4, и ещё две от 5 до 6.

Мак­си­маль­ное число, под­хо­дя­щее под все эти усло­вия — 96644.

**7.** 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 11.

|  |
| --- |
|  |
| var x, a, b: integer;begin    readln(x);    a := 0; b := 1;    while x > 0 do    begin        a := a+1;        b := b\*(x mod 1000);        x := x div 1000;    end;    writeln(a); write(b);end. |

**По­яс­не­ние.**

За­ме­тим, что так как a = 2, то .

Всего две ите­ра­ции цикла. Так как 11 — про­стое число, то в одной ите­ра­ции x mod 1000 долж­но быть равно 11, а в дру­гой — 1. Число по­лу­чит­ся мень­ше, если 11 будет по­лу­че­но на пер­вой ите­ра­ции.
По­лу­ча­ем число 1011.

**8.** 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 2, а потом 11.

|  |
| --- |
|  var x, a, b: integer;begin    readln(x);    a := 0; b := 1;    while x > 0 do    begin        a := a+1;        b := b\*(x mod 1000);        x := x div 1000;    end;    writeln(a); write(b);end. |

**По­яс­не­ние.**

За­ме­тим, что так как a = 2, то .

Всего две ите­ра­ции цикла. Так как 11 — про­стое число, то в одной ите­ра­ции x mod 1000 долж­но быть равно 11, а в дру­гой — 1. Число по­лу­чит­ся боль­ше, если 11 будет по­лу­че­но на вто­рой ите­ра­ции. По­лу­ча­ем число 11001.

**9.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­боль­шее 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 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 then begin

  M:= (x mod 10) \* 2;

end;

 Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 10 это по­след­няя цифра *x*. Чтобы M при­ня­ло зна­че­ние M = 10 в числе x долж­но при­сут­ство­вать цифра 5.

Т. к. мы ищем наи­боль­шее *x*, сде­ла­ем первую цифру ис­ход­но­го числа, рав­ной 9, при этом если вто­рая цифра 5, тогда на тре­тьем шаге усло­вие M < *x* не вы­пол­ня­ет­ся (10>9), и на экра­не мы по­лу­чим нуж­ное нам число.

  Оста­лось опре­де­лить по­след­нюю цифру ис­ход­но­го числа *x*. После пер­во­го шага *x* = 95, но мак­си­маль­ный оста­ток от де­ле­ния на 10 равен 9, сле­до­ва­тель­но, M после пер­во­го шага не может пре­вы­сить 9 \* 2 = 18, а 18 < 95, зна­чит будет вы­пол­нять­ся нуж­ный нам вто­рой шаг.

 Ис­ко­мое число 959.

Ответ: 959.

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

 Если b = *x*(1) \* *x*(2) \* *x*(3) = 0, где число в скоб­ках озна­ча­ет номер цифры в числе *x*, то хотя бы один из мно­жи­те­лей равен нулю, но это озна­ча­ет, что на каком-то шаге *x* на­це­ло де­лит­ся на 10. Если это про­ис­хо­дит на пер­вом шаге, то *x*(1) и *x*(2) могут быть лю­бы­ми од­но­знач­ны­ми чис­ла­ми. Т. к. мы хотим по­лу­чить наи­боль­шее число, то *x* = 990.

Ответ: 990.

**11.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­боль­шее 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, т. к. на вто­ром шаге число *x* дву­знач­ное, а од­но­знач­ное число все­гда мень­ше лю­бо­го дву­знач­но­го, а зна­чит, M может при­об­ре­сти ин­те­ре­су­ю­щее нас зна­че­ние 7. По­это­му тре­тья цифра числа *x* *x*(3) = 9.

 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 наи­боль­шее число *x*, мы берём *x*(1) = 7. Оста­лось верно опре­де­лить *x*(2).

  Число *x* до­стиг­нет мак­си­му­ма при *x*(2)=7, при этом на тре­тьем шаге пе­ре­мен­ная M уже не из­ме­нит­ся, по­то­му что на­ру­ше­но усло­вие M < x (7 = 7).

Окон­ча­тель­но по­лу­ча­ем: *x* = 779. Ответ: 779.

**12.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *a* и *b*. Ука­жи­те наи­боль­шее 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 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 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 мак­си­маль­ное число: 711  Ответ: 711.

**13.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­боль­шее 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 = 0 then

            M:= M + (x mod 10) div 2;

        x:= x div 10;

    end;     writeln(L); write(M); end.

**По­яс­не­ние.**

На­пом­ним:

X div a - целая часть от де­ле­ния числа X на a,

X mod a - оста­ток от де­ле­ния числа X на a.

 Пока X боль­ше нуля вы­пол­ня­ет­ся уве­ли­че­ние L на еди­ни­цу. Если X чет­ное (оста­ток от де­ле­ния на 2 равен 0), то M уве­ли­чи­ва­ет­ся на по­ло­ви­ну цифры, сто­я­щей в еди­ни­цах числа X, X со­кра­ща­ет­ся на один раз­ряд спра­ва.

L - ко­ли­че­ство цифр в числе X,

M - сумма, ко­то­рая на­хо­дит­ся таким об­ра­зом:

L=3, сле­до­ва­тель­но X = a \* 100 + b \* 10 + c,

M = (a + b + c) / 2, если все числа четны. Если хоть одно из чисел не­чет­ное, то сумма будет со­сто­ять из 2-х цифр, так как при М = 7, a + b + c = 14, т. е. нужны как ми­ни­мум 2 цифры чтобы на­брать дан­ную сумму.

 Чтобы по­до­брать мак­си­маль­ное число при­мем, что a = 9, b + c = 14, и обе цифры чет­ные, так что b = 8 — наи­боль­шая чет­ная цифра, c = 6.

Пра­виль­ный ответ: 986.

**14.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­боль­шее 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*.

 Пред­ста­вим 7 в виде суммы: 7 = 7 + 0 + 0 = 1 + 6 + 0 = 1 + 5 + 1 = 1 + 4 + 2 = 1 + 3 + 3 и т. д. Видно, что мы не можем ис­поль­зо­вать цифры боль­ше 7, по­это­му для по­лу­че­ния мак­си­маль­но­го числа ис­поль­зу­ем пред­став­ле­ние 7 = 7 + 0 + 0.

Из этих цифр со­ста­вим наи­боль­шее число: 700.

Ответ: 700.

**15.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­боль­шее 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, и по­сколь­ку мы ищем мак­си­маль­ное число, то усло­вие M < x будет вы­пол­нять­ся. Од­на­ко, мы можем на­ру­шить усло­вие нечётно­сти (x mod 2 = 1), если сде­ла­ем вто­рую цифру чётной, тогда M далее не из­ме­нит­ся, по­то­му что на тре­тьем шаге оста­нет­ся толь­ко пер­вая цифра числа *x*, а любое од­но­знач­ное число мень­ше любго дву­знач­но­го, вклю­чая 10.

 По­это­му пер­вая цифра может быть любой, по­ло­жим её рав­ной 9, вто­рая цифра долж­на быть чётной — вы­би­ра­ем наи­боль­шую, т. е. 8. В ре­зуль­та­те по­лу­ча­ем число 985.

Ответ: 985.

**16.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­боль­шее 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 наи­боль­шее зна­че­ние вво­ди­мо­го числа *x* = 998.

Ответ: 998.

**17.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­боль­шее 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, по­сколь­ку есть усло­вие x mod 2 = 1.

Пред­ста­вим 7 в виде: 7 = 4 + 3.

Для по­лу­че­ния наи­боль­ше­го *x* по­ло­жим первую цифру *x*(1) = 9, она даст вклад 4 на тре­тьем шаге цикла, тогда дру­гая цифра числа *x* долж­на быть раной 7, она даст вклад 3 на вто­ром или на пер­вом шаге цикла. Остав­шу­ю­ся цифру мы долж­ны вы­брать чётной, а наи­боль­шая чётная цифра есть 8.

Те­перь со­став­ля­ем из этих цифр наи­боль­шее число: *x* = 987.

Ответ: 987.

**18.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­боль­шее 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 = 1 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 = 1 then

M:= M + (x mod 10);

end;

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

За­ме­тим, что x mod 10 при­ни­ма­ет зна­че­ния от 0 до 9, но мы хотим по­лу­чить сумму остат­ков от трёх де­ле­ний 8, по­это­му брать в ка­че­стве цифр числа *x* цифру 9 мы не можем.

 Чтобы М уве­ли­чи­ва­лось число *x* долж­но со­дер­жать нечётные цифры (об этом го­во­рит усло­вие x mod 2 = 1). Чтобы их отыс­кать пред­ста­вим 8 как сумму нечётных чисел: 8 = 7 + 1 = 5 + 3. Дру­гих ва­ри­ан­тов пред­став­ле­ния у нас нет. Зна­чит, нужно, чтобы две цифры были нечётными, и одна чётной.

 Мы хотим по­лу­чить мак­си­маль­ное число, по­это­му по­ста­вим на пер­вое место 8, т. к. 8 боль­ше лю­бо­го нечётного числа, кроме 9. Тогда из пред­став­ле­ния числа 8 в виде суммы найдём и две дру­гие цифры. Из цифр 8, 7, 1 и 8, 5, 3 можно со­ста­вить числа и вы­брать наи­боль­шее, таким будет число 871.

Ответ: 871.

**19.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­боль­шее 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 мак­си­ма­льое *x*, по­это­му ис­ко­мое число 999.

Ответ: 999.

**20.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число *x* , эта про­грам­ма пе­ча­та­ет два числа, *L* и *M*. Ука­жи­те наи­боль­шее 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 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. Тогда наи­боль­шее зна­че­ние вво­ди­мо­го числа x = 998.

Ответ: 998.

**21.** 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 на­ча­ла 2, а потом 14.

|  |
| --- |
| var x, a, b : integer;beginreadln(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, мак­си­маль­ное число *x* = 72.

Ответ: 72.

**22.** 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 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 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 72 в виде: 72 = 9 \* 8. Дру­го­го пред­став­ле­ния, в ко­то­ром оба мно­жи­те­ля од­но­знач­ные число 72 не имеет. Сле­до­ва­тель­но, мак­си­маль­ное число *x* = 98.

Ответ: 98.

**23.** 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 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 наи­боль­шее *x* по­ло­жим его первую цифру рав­ной *x*(1) = 9. Тогда вто­рая цифра *x*(2) будет равна 10 - 9 = 1, сле­до­ва­тель­но ниа­боль­шее число *x* = 91.

Ответ: 91.

**24.** Ниже за­пи­са­на про­грам­ма. 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 наи­боль­шее из таких чисел x, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, а потом 5.

**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*.

Число 5 про­стое, по­это­му 5 = 5 \* 1 \* 1  — един­ствен­ное пред­став­ле­ние числа 5. Сле­до­ва­тель­но, мак­си­маль­ное число *x* = 511.

Ответ: 511.

**25.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число x, этот ал­го­ритм пе­ча­та­ет два числа *a* и *b*.

|  |
| --- |
| 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. |

Ука­жи­те наи­боль­шее из таких чисел x, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 8.

**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 **двух­знач­ным**.

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*.

Число 8 можно пред­ста­вить в виде суммы 8 + 0 = 7 +1 = 6 + 2 = 5 + 3 = 4 + 4. Нас ин­те­ре­су­ет наи­боль­шее число, по­это­му *x* = 80. Ответ: 80.

**26.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число х, этот ал­го­ритм пе­ча­та­ет два числа а и Ь. Ука­жи­те наи­боль­шее из таких чисел х, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 35.

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

У числа 35 су­ще­ству­ют два де­ли­те­ля, мень­ших 10, 7 и 5. Сле­до­ва­тель­но, су­ще­ству­ют два x, удо­вле­тво­ря­ю­щие за­дан­ным a и b: 75 и 57. В за­да­нии ука­за­но найти наи­боль­ший x, сле­до­ва­тель­но, пра­виль­ный ответ — 75.

**27.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число х, этот ал­го­ритм пе­ча­та­ет два числа а и Ь. Ука­жи­те наи­боль­шее из таких чисел х, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 21.

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

У числа 21 су­ще­ству­ют два де­ли­те­ля, мень­ших 10, 3 и 7. Сле­до­ва­тель­но, су­ще­ству­ют два x, удо­вле­тво­ря­ю­щие за­дан­ным a и b: 73 и 37. В за­да­нии ука­за­но найти наи­боль­ший x, сле­до­ва­тель­но, пра­виль­ный ответ — 73.

**28.** Ниже за­пи­сан ал­го­ритм. После вы­пол­не­ния ал­го­рит­ма было на­пе­ча­та­но 3 числа. Пер­вые два на­пе­ча­тан­ных числа - это числа 7 и 42. Какое наи­боль­шее число может быть на­пе­ча­та­но тре­тьим?

var x, у, z: integer;

var r, a, b: integer;

begin

    readln(x, у);

    if у >x then begin

        z:= x;x:= y;y:= z;

    end;

    a:= x; b:= у;

    while b>0 do begin

        r := a mod b;

        a := b;

        b := r;

    end;

    writeln(a);

    writeln(x);

    write(y);

end.

**По­яс­не­ние.**

Сна­ча­ла вво­дят­ся два числа и пе­ре­став­ля­ют­ся так, чтобы в пе­ре­мен­ной x было наи­боль­шее число, а в пе­ре­мен­ной y – наи­мень­шее из двух:

"if у >x then begin

z:= x;x:= y;y:= z;"

"a:= x; b:= у;

while b>0 do begin

r := a mod b;

a := b;

b := r; "

Ал­го­ритм по­ис­ка наи­боль­ше­го об­ще­го де­ли­те­ля, ко­то­рый в итоге ока­зы­ва­ет­ся в пе­ре­мен­ной а.

На экран вы­во­дит­ся: сна­ча­ла зна­че­ние пе­ре­мен­ной a (наи­боль­ший общий де­ли­тель ис­ход­ных чисел), затем зна­че­ние x (боль­шее из ис­ход­ных чисел) и зна­че­ние y (мень­шее из ис­ход­ных чисел).

По усло­вию пер­вое число — 7, вто­рое — 42. Сле­до­ва­тель­но, ис­ко­мое число долж­но де­лить­ся на 7 и быть мень­ше 42. Наи­боль­шее из таких чисел — 35.

**29.** Ниже за­пи­сан ал­го­ритм. После вы­пол­не­ния ал­го­рит­ма было на­пе­ча­та­но 3 числа. Пер­вые два на­пе­ча­тан­ных числа - это числа 11 и 66. Какое наи­боль­шее число может быть на­пе­ча­та­но тре­тьим?

var x, у, z:= integer;

var r, a, b:= integer;

begin

    readln(x, у);

    if у >x then begin

        z:= x;x:= y;y:= z;

    end;

    a:= x; b:= у;

    while b>0 do begin

        r := a mod b;

        a := b;

        b := r;

    end;

    writeln(a);

     writeln(x);

     write(y);

end.

**По­яс­не­ние.**

Сна­ча­ла вво­дят­ся два числа и пе­ре­став­ля­ют­ся так, чтобы в пе­ре­мен­ной x было наи­боль­шее число, а в пе­ре­мен­ной y – наи­мень­шее из двух:

"if у >x then begin

z:= x;x:= y;y:= z;"

"a:= x; b:= у;

while b>0 do begin

r := a mod b;

a := b;

b := r; "

Ал­го­ритм по­ис­ка наи­боль­ше­го об­ще­го де­ли­те­ля, ко­то­рый в итоге ока­зы­ва­ет­ся в пе­ре­мен­ной а.

На экран вы­во­дит­ся: сна­ча­ла зна­че­ние пе­ре­мен­ной a (наи­боль­ший общий де­ли­тель ис­ход­ных чисел), затем зна­че­ние x (боль­шее из ис­ход­ных чисел) и зна­че­ние y (мень­шее из ис­ход­ных чисел).

По усло­вию пер­вое число — 11, вто­рое — 66. Сле­до­ва­тель­но, ис­ко­мое число долж­но де­лить­ся на 11 и быть мень­ше 66. Наи­боль­шее из таких чисел — 55.

**30.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число х, этот ал­го­ритм пе­ча­та­ет два числа а и b.
Ука­жи­те наи­боль­шее из таких чисел x, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 13.

|  |
| --- |
| var х, a, b: integer; begin    readln(х);     а : = 0; b : = 0;     while х > 0 do         begin            а := а+1;            b := b + (х mod 100) ;             х := х 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 остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 100 это рав­но­силь­но от­се­че­нию по­след­них двух цифр.

Из при­ве­ден­но­го цикла видно, что 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; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко пар цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a сов­па­да­ет с чис­лом пар цифр в *x*. Для того, чтобы a стало a = 2, *x* долж­но быть **четырёхзнач­ным**.

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

b := b + (х mod 100);

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 100 это две по­след­ние цифры *x*; сле­до­ва­тель­но, число b по­лу­ча­ет­ся сум­мой пар цифр числа *x*.

Таким об­ра­зом b = *x*(1) + *x*(2) = 13, где число в скоб­ках озна­ча­ет дву­знач­ную цифру в за­пи­си числа *x*.

Наи­боль­шее число, удо­вле­тво­ря­ю­щее дан­ным усло­ви­ям — 1300.

Ответ: 1300.

**31.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число х, этот ал­го­ритм пе­ча­та­ет два числа а и b. Ука­жи­те наи­боль­шее из таких чисел x, при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 2, а потом 17.

|  |
| --- |
| var х, a, b: integer; begin    readln(х);     а : = 0; b : = 0;     while х > 0 do         begin            а := а+1;            b := b + (х mod 100);             х := х 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 остав­ля­ет толь­ко целую часть от де­ле­ния, то при де­ле­нии на 100 это рав­но­силь­но от­се­че­нию по­след­них двух цифр.

Из при­ве­ден­но­го цикла видно, что 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; по­это­му цикл вы­пол­ня­ет­ся столь­ко раз, сколь­ко пар цифр в де­ся­тич­ной за­пи­си вве­ден­но­го числа, при этом число a столь­ко же раз уве­ли­чи­ва­ет­ся на 1. Сле­до­ва­тель­но, ко­неч­ное зна­че­ние a будет рав­нять­ся двум, если число на­хо­дит­ся в диа­па­зо­не [100; 10 000).

Те­перь рас­смот­рим опе­ра­тор из­ме­не­ния b:

while x>0 do begin

b := b + (х mod 100);

end;

Опе­ра­тор mod остав­ля­ет толь­ко оста­ток от де­ле­ния, при де­ле­нии на 100 это две по­след­ние цифры *x*; сле­до­ва­тель­но, число b по­лу­ча­ет­ся сум­мой пар цифр числа *x*.

Таким об­ра­зом b = *x*(1) + *x*(2) = 17, где число в скоб­ках озна­ча­ет дву­знач­ную цифру в за­пи­си числа *x*.

Наи­боль­шее число, удо­вле­тво­ря­ю­щее дан­ным усло­ви­ям — 1700.

Ответ: 1700.

**32.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число , этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­боль­шее из таких чисел , 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        c := 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. Мак­си­маль­ным чис­лом, удо­вле­тво­ря­ю­щим дан­ным усло­ви­ям яв­ля­ет­ся число 99888. Ответ: 99888.

**33.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число , этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­боль­шее из таких чисел , 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        c := 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. Мак­си­маль­ным чис­лом, удо­вле­тво­ря­ю­щим дан­ным усло­ви­ям яв­ля­ет­ся число 99988. Ответ: 99988.

**34.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число , этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­боль­шее из таких чисел , при вводе ко­то­рых ал­го­ритм пе­ча­та­ет сна­ча­ла 3, 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        c := 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 3, не­об­хо­ди­мо, чтобы число *x* имело шесть раз­ря­дов. Три цифры в этом числе долж­ны быть чётные, а три —нечётные. Мак­си­маль­ным чис­лом, удо­вле­тво­ря­ю­щим дан­ным усло­ви­ям яв­ля­ет­ся число 999888.

Ответ: 999888.

**35.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число , этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­боль­шее из таких чисел , 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        c := 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. Мак­си­маль­ным чис­лом, удо­вле­тво­ря­ю­щим дан­ным усло­ви­ям яв­ля­ет­ся число 99888.

Ответ: 99888.

**36.** Ниже за­пи­са­на про­грам­ма. По­лу­чив на вход число , этот ал­го­ритм пе­ча­та­ет два числа: *a* и *b*. Ука­жи­те наи­боль­шее из таких чисел , 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        c := 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. Мак­си­маль­ным чис­лом, удо­вле­тво­ря­ю­щим дан­ным усло­ви­ям яв­ля­ет­ся число 99988.

Ответ: 99988.

**37.** 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 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 равно 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. Наи­боль­шее *x*, име­ю­щее сумму цифр 25 и удо­вле­тво­ря­ю­щее усло­ви­ям за­да­чи равно 997.

Ответ: 997.

**38.** 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 2, а потом 11.

|  |
| --- |
|  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 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, в пе­ре­мен­ной *b* на­кап­ли­ва­ет­ся сумма цифр числа *x*. Наи­боль­шее *x* удо­вле­тво­ря­ю­щее усло­ви­ям за­да­чи равно 92.

Ответ: 92.

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

|  |
| --- |
| program B08;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*. Наи­боль­шее *x* удо­вле­тво­ря­ю­щее усло­ви­ям за­да­чи равно 90. Ответ: 90.

**40.** 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 2, а потом 16.

|  |
| --- |
| 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 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, в пе­ре­мен­ной *b* на­кап­ли­ва­ет­ся сумма цифр числа *x*. Наи­боль­шее *x* удо­вле­тво­ря­ю­щее усло­ви­ям за­да­чи равно 97.

Ответ: 97.

**41.** 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 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;

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, в пе­ре­мен­ной *b* на­кап­ли­ва­ет­ся сумма цифр числа *x*. Наи­боль­шее *x* удо­вле­тво­ря­ю­щее усло­ви­ям за­да­чи равно 90. Ответ: 90.

**42.** 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 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 100, чтобы числа L = x − 30 и M = x + 30 имели НОД рав­ный 60. При x = 150 L = 120 и M = 180, наи­боль­ший общий де­ли­тель равен 60. При 100 < x < 150, наи­боль­ший общий де­ли­тель L и M не может быть равен 60.

Ответ: 150.

**43.** 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 60.

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.**

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 100, чтобы числа L = x − 30 и M = x + 30 имели НОД рав­ный 60. При x = 150 L = 120 и M = 180, наи­боль­ший общий де­ли­тель равен 60. При 100 < x < 150, наи­боль­ший общий де­ли­тель L и M не может быть равен 60.

Ответ: 150.