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 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.

2) **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 (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.

**3) 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 = 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.