Вспомогательный алгоритм

Определите, какое число будет напечатано в результате выполнения следующего алгоритма:

```
Var a,b,t,M,R :integer;
Function F(x:integer):integer;
begin
   F:=(x+5)*(x+3);
end;
BEGIN
   a:= -5; b:=5;
   M:=a; R:=F(a);
   for t:=a to b do begin
      if (F(t)> R)then begin
         M:=t;
         R:=F(t);
      end;
   end;
   write(R);
END.
```

1. Алгоритм предназначен для поиска наибольшего значения функции F(t) на отрезке от a до b.
2. Квадратный трехчлен F(t) положительным старшим коэффициентом пересекает ось абсцисс в точках -5 и -3 и, следовательно, возрастет на луче [-3; ∞). Поэтому наибольшее значение функции достигается в точке 5 и равно F(5)=80.

Найдите число различных значений входной переменной **k**, при которых программа выдаёт тот же ответ, что и при входном значении **k = 55**. Значение **k = 55** также включается в подсчёт различных значений **k**.

```
var k, i : longint;
function f(n: longint):longint;
begin
f := 3*n*n+1
end;
begin
 readln(k);
 i := 0;
while (f(i)<k) do
 i := i+1;
 writeln(i)
end.
```

Программа выводит минимальное **i** , которое удовлетворяет неравенству **$3*i^2>=K$** . Для **k = 55** **i=5**

Найдём все подходящие **k** . Это те **k**, для которых выполнены оба неравенства $K > 3 * 4^2 + 1$ и $K \leq 3 * 5^2 + 1$

Таким образом получаем 49 <K <=76. Этот промежуток содержит 27 целых значений .

Вспомогательный алгоритм *- это алгоритм, оформленный так, что он может вызываться и использоваться в другом алгоритме.*

Подпрограмма *- это фрагмент программы, реализующий вспомогательный алгоритм, к которому можно обратиться из любого места программы любое число раз.*

Подпрограмма

процедуры

функции

В процедуру могут передаваться параметры, то есть некоторые переменные, которые могут использоваться внутри процедуры. При вызове процедуры с помощью оператора вызова этим переменным присваиваются значения, указанные в этом операторе. Параметры, описанные в заголовке процедуры, называются *формальными значения*.
Значения, которые присваиваются этим параметрам в процессе вызова — *фактическими параметрами*.

функция через свое имя возвращает одно значение определенного типа и может, использоваться в выражениях наряду со встроенными функциями

Процедура

Процедура оформляется следующим образом:	
Алгоритмический язык	**Паскаль**
алг <имя процедуры> (<список параметров>) <операторы> кон	procedure <имя процедуры> (<список параметров>); <описание> begin <операторы> end

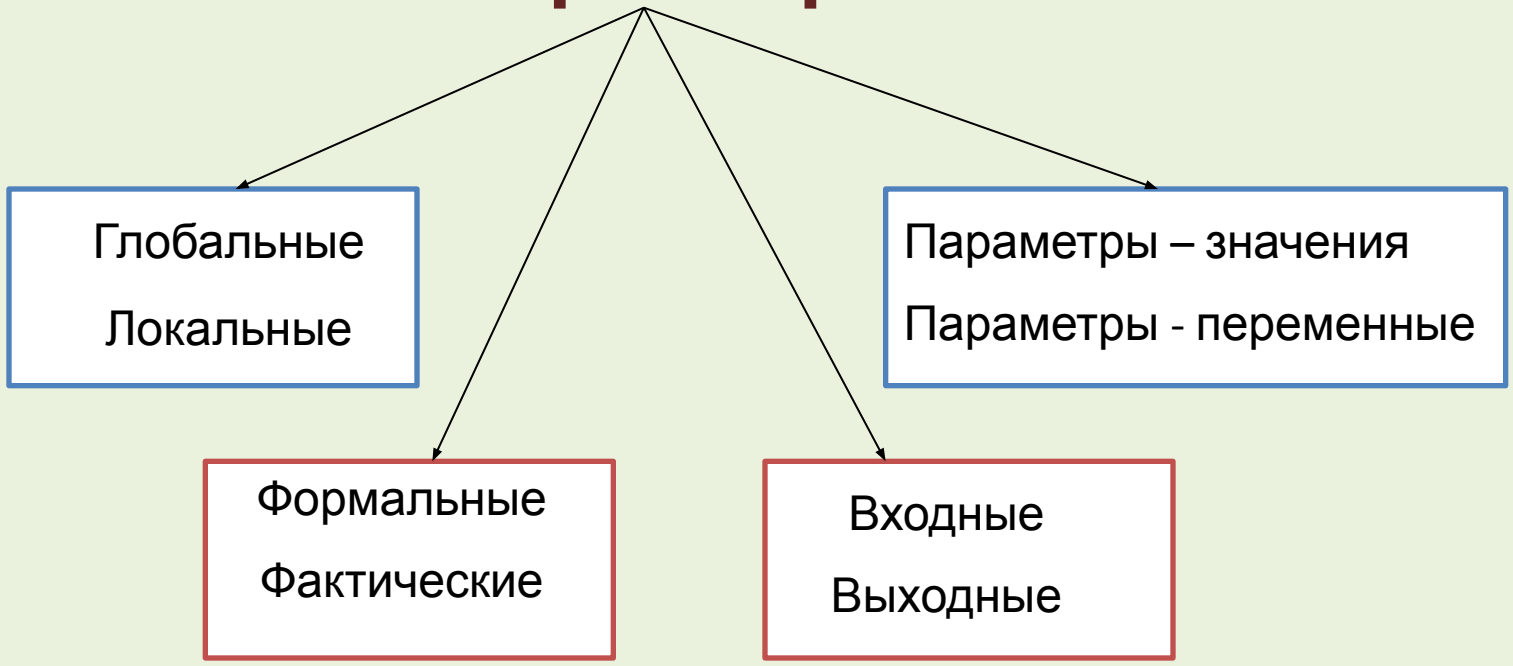
```
procedure SubTest(a,b:integer; var c:real, var d:integer);
```

Функция

Бейсик	Паскаль
FUNCTION <имя>(<параметры>) <операторы> END FUNCTION	function <имя>(<параметры>):<тип результата>; <описания> begin <операторы> end

<Переменная> := <Функция> (<Параметры>)

Параметры


Глобальные
Локальные
Параметры – значения
Параметры - переменные
Формальные
Фактические
Входные
Выходные

Параметры

- *Глобальные* – описываются в головном модуле, доступны любой подпрограмме.

- *Локальные* – используются только в процедуре, они могут быть или не быть, описываются после слова VAR, с указанием типа.

Параметры

- *Формальные* – описываются в заголовке процедуры, к ним относятся входные и выходные параметры.
- *Входные* – это и параметры значений, описываются через запятую с указанием типа. При выходе из процедуры – не сохраняются.
- *Выходные* – это и параметры переменные, описываются после VAR через запятую, с указанием типа. При выходе из процедуры – сохраняются.
- *Фактические* – располагаются в головном модуле при вызове процедуры.

Формальные и *фактические* параметры должны совпадать по 3 признакам:

- ***по количеству***
- ***по типу***
- ***по порядку следования***

Общий вид структуры подпрограммы

Procedure <имя> (формальные параметры);

VAR (описание локальных параметров, они могут быть или не быть)

begin

тело процедуры

end; (конец процедуры)

BEGIN (основная программа)

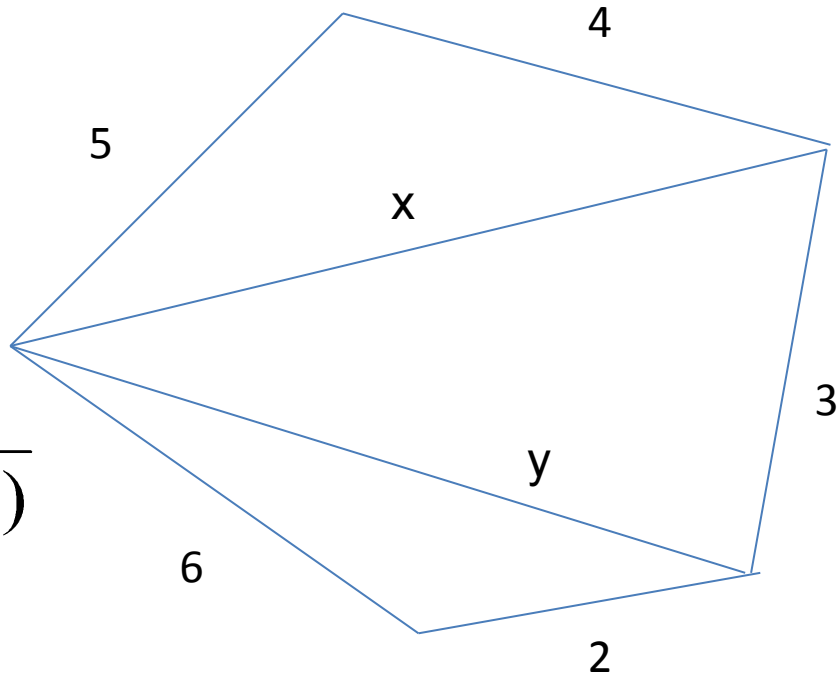
Фактические параметры

END.

Найти площадь фигуры:

$$P = \frac{a+b+c}{2}$$

$$s = \sqrt{p \cdot (p-a) \cdot (p-b) \cdot (p-c)}$$



Procedure treugolnik (a, b, c);

VAR a, b, c: integer;

begin

$$P = \frac{a+b+c}{2}$$

$$s = \sqrt{p \cdot (p-a) \cdot (p-b) \cdot (p-c)}$$

end;

Рассмотрим использование процедуры на примере программы поиска наибольшего из двух целых чисел

```
var x,y,m: integer;
procedure MaxNum(a,b: integer; var max: integer);
begin
if a>b then max:=a else max:=b;
end;
begin
writeln('Введите x,y ');
readln(x,y);
MaxNum(x,y,m);
writeln('Наибольшее число: ',m);
end.
```

Задача с использованием функции :

```
var x,y,m: integer;
function MaxNum(a,b: integer): integer;
var max: integer;
begin
if a>b then max:=a else max:=b;
MaxNum:= max;
end;
begin
writeln('Введите x,y ');
readln(x,y);
m := MaxNum (x,y);
writeln('Наибольшее число: ',m);
end.
```