

## Лабораторная работа №2

Тема: *Разработка и реализация программ с разветвленной структурой*

### Цель работы:

получить навыки составления и реализации программ с разветвленной структурой, используя средства языка C/C++;

### Задания к лабораторной работе

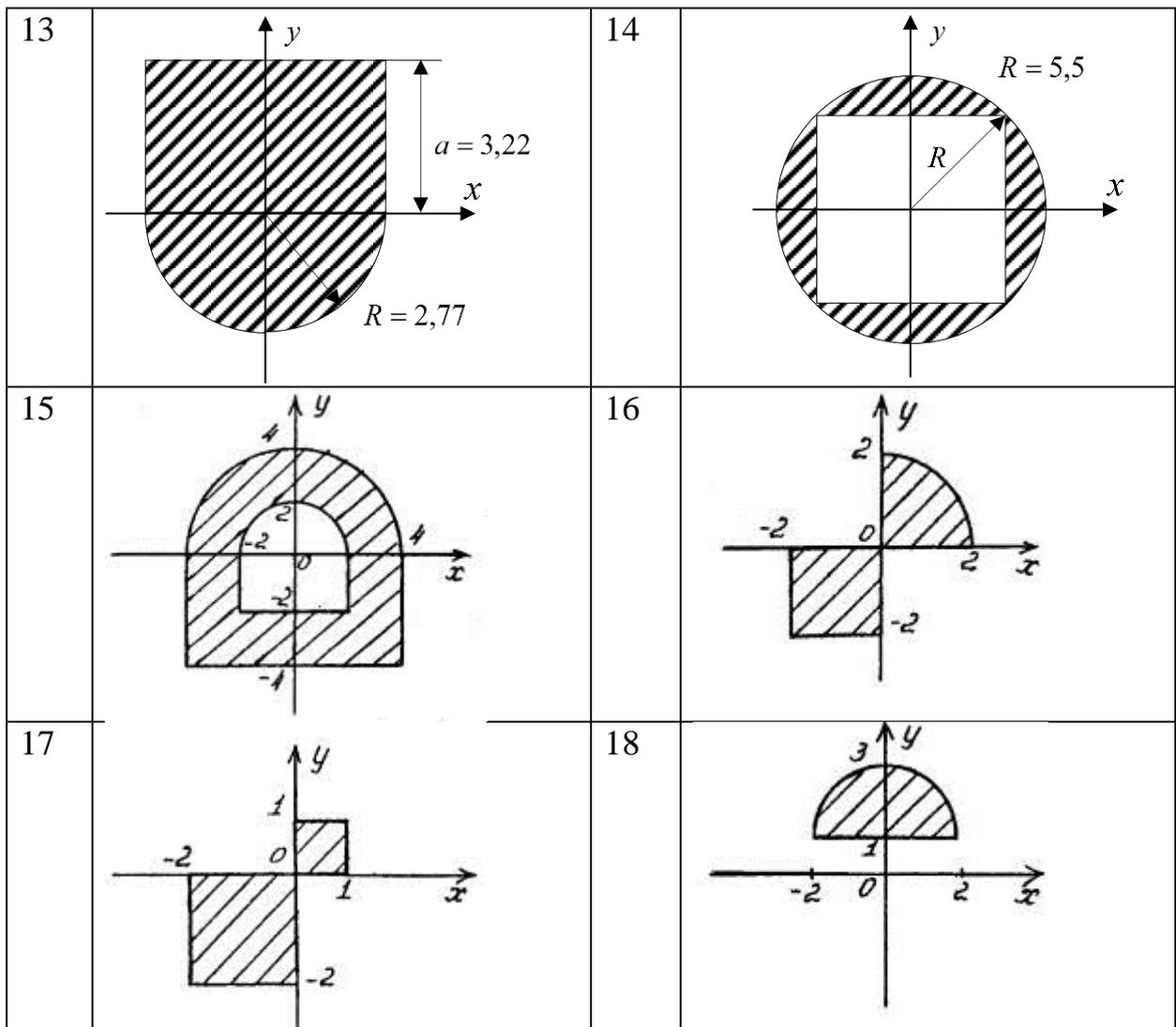
**Задание 2.1** Представить математическую запись фрагмента программы и вычислить значение переменной X после его выполнения. Обозначение: N - это номер варианта.

№вар	Фрагмент программы	№вар	Фрагмент программы
1-2	t=17*n; x=t; if (t<10    t>30) x=3; else if (t<=20) x=0;	3-4	t=n; x=0; if (t<0)x= -t; else x= t;
5-6	a=n; b=13; c=12; x=a; if (x<b) x=b; if (x<c) x=c;	7-8	a=n; b=17; c=18; x=a; if (b<x) x=b; if (c<x) x=c;
9-10	x=n; y=0; if (x>10) y=x*x-n; if (x<10) y=x;	11-12	x=n; y=x%4; if(x>1 && x<3) y=9; if (x<=1) y=x;
13-14	x=n; y=x; if(x>0 && x<10) y=1; if (x>=10) y=1/(exp(x)-1);	15-16	x=-7; t=pow(x, n); if (t>0)x=pow(t, 1.0/3); else x= t*t*t;

**Задание 2.1** Написать программу, которая выводит на экран значение *true*, если точка A с координатами x, y принадлежит заштрихованной области, и *false* в противном случае.

№ вар.	Область	№ вар.	Область
1		2	

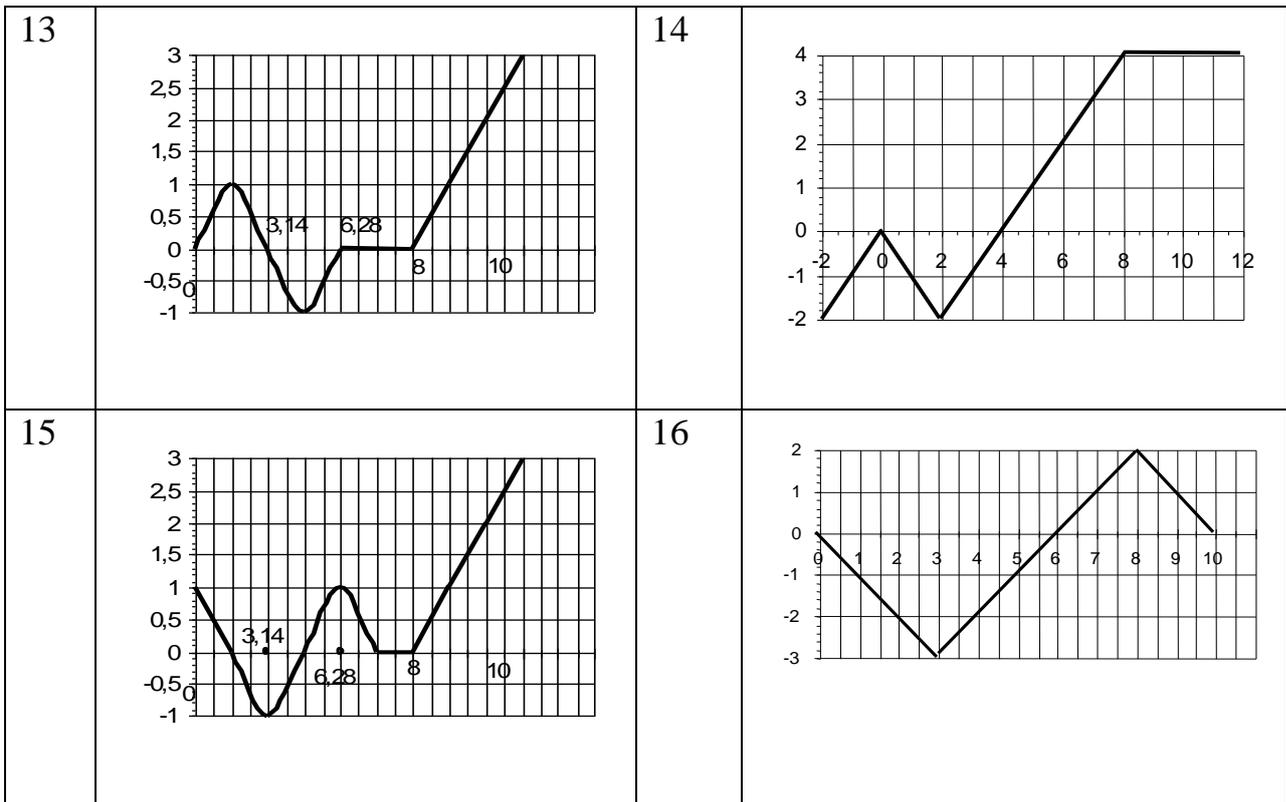
3		4	
5		6	
7		8	
9		10	
11		12	



Задание 3 Составить программу вычисления значения функции, заданной графически. (Значение аргумента  $X$  вводится произвольно с клавиатуры).

№ вар.	Функция	№ вар.	Функция
1		2	

3		4	
5		6	
7		8	
9		10	
11		12	



Додаткові завдання

<http://www.e-olimp.com.ua/problems/206>

<http://www.e-olimp.com.ua/problems/407>

<http://www.e-olimp.com.ua/problems/1312>

<http://www.e-olimp.com.ua/problems/1622>

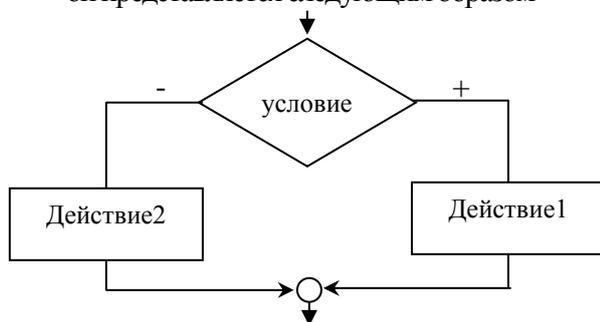
<http://www.e-olimp.com.ua/problems/2861>

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ №2**  
**ТЕМА: РАЗРАБОТКА И РЕАЛИЗАЦИЯ ПРОГРАММ С РАЗВЕТВЛЕННОЙ**  
**СТРУКТУРОЙ**

Представить математическую запись фрагмента программы и вычислить значение переменной X после его выполнения. Обозначение: N - это номер варианта

```
t=n; x=0;  
if (t<0) x=(t-1);  
else x= t;
```

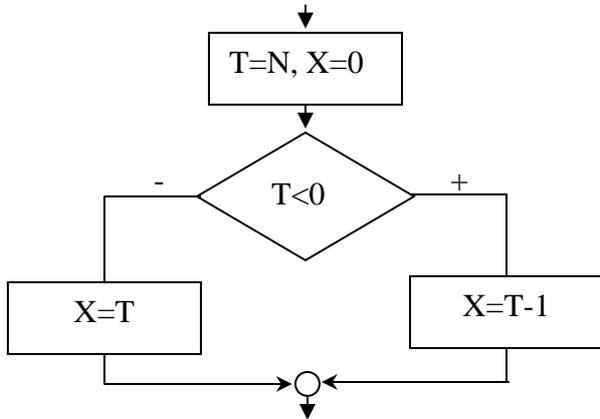
В данном случае рассматривается полный условный оператор IF. На блок-схеме в общем виде он представляется следующим образом



На C/C++ этот фрагмент блок-схемы записывается следующим образом:

```
If (условие) действие1;
    else действие2;
```

Если условие истинно, то выполняется действие1, иначе (если условие ложно) выполняется действие2. Таким образом, блок-схема решаемого примера



Пусть  $N = 5$ . Тогда в соответствии с блок-схемой  $T=5, X=0$ .

Проверяем условие.  $T$  в данном случае больше 0, то есть условие  $T < 0$  ложно. Движемся по левой ветке.  $X=T$ , то есть  $X=5$ . Окончательный ответ после выполнения фрагмента программы  $X=5$ .

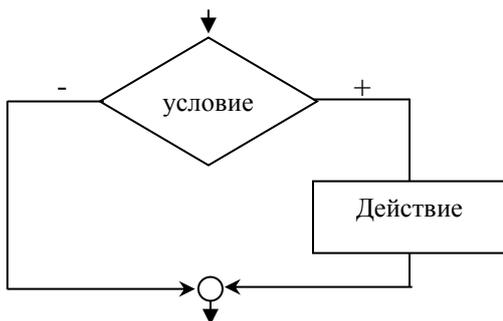
Пусть  $N = -2$ . Тогда в соответствии с блок-схемой  $T=-2, X=0$ . Проверяем условие.  $T$  в данном случае меньше 0, то есть условие  $T < 0$  истинно. Движемся по правой ветке.  $X=T-1$ , то есть  $X = -2-1 = -3$ . Окончательный ответ после выполнения фрагмента программы  $X = -3$ .

Вывод:  $x(t) = \begin{cases} t - 1, & \text{при } t < 0 \\ t, & \text{при } t \geq 0 \end{cases}$

Рассмотрим еще один пример. В случае отсутствия действия, которое нужно выполнить в случае, когда условие ложно оператор `if` называется неполным и имеет вид

```
If (условие) действие;
```

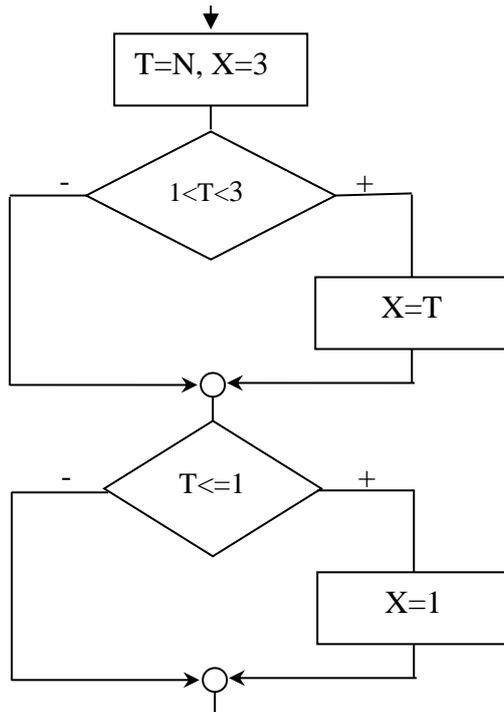
Блок-схема такого варианта представлена ниже



Рассмотрим пример

```
t=n; x=3;
if (t>1 && t<3) x=t;
if (t<=1) x=1
```

Блок-схема этого примера



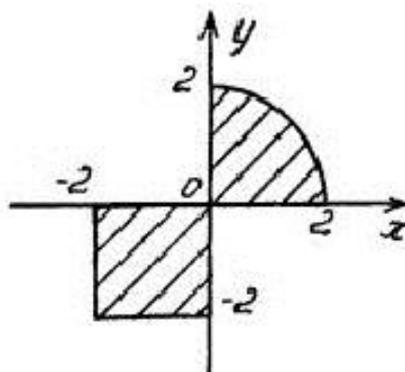
Пусть  $N=5$ , тогда  $T=5$ ,  $X=3$ . Условие  $1 < T < 3$  не выполняется, движемся по левой ветке, приходим к условию  $T \leq 1$ . Это условие тоже не выполняется. Значит после выполнения фрагмента программы  $X=3$ .

Пусть  $N=-1$ , тогда  $T=-1$ ,  $X=-1$ . Условие  $1 < T < 3$  не выполняется, движемся по левой ветке, приходим к условию  $T \leq 1$ . Это условие выполняется. Значит, движемся по правой ветке и после выполнения фрагмента программы  $X=1$ .

Вывод :

$$X = \begin{cases} T, & \text{если } 1 < T < 3 \\ 1, & \text{если } T \leq 1 \\ 3, & \text{в остал. случаях} \end{cases}$$

*Задание 2.1.* Написать программу, которая выводит на экран значение *true*, если точка  $A$  с координатами  $x$ ,  $y$  принадлежит заштрихованной области, и *false* в противном случае.



Данная часть плоскости состоит из двух непересекающихся областей. Таким образом если точка попадет в одну область или в другую она будет принадлежать заданной плоскости.

Область находящаяся в первой четверти ограничена тремя линиями:  $x=0$ ,  $y=0$ ,  $x^2+y^2=2^2$ , при чем область находится выше  $y=0$ , включая саму линию, то есть  $y \geq 0$ ; справа от  $x=0$ , включая саму линию, то есть  $x \geq 0$ , и внутри окружности  $x^2+y^2=2^2$ , то есть  $x^2+y^2 \leq 2^2$

Таким образом, логическое выражение для этой области  $x \geq 0 \wedge y \geq 0 \wedge x^2+y^2 \leq 2^2$ .

Область находящаяся во второй четверти ограничена линиями:  $x=0$ ,  $y=0$ ,  $x=-2$ ,  $y=-2$ , при чем область находится ниже  $y=0$ , то есть  $y \leq 0$ ; слева от  $x=0$ , то есть  $x \leq 0$ , выше  $y=-2$ , то есть  $y \geq -2$ ; справа от  $x=-2$ , то есть  $x \geq -2$ .

Таким образом, логическое выражение для второй области  $x \geq (-2) \wedge x \leq 0 \wedge y \geq (-2) \wedge y \leq 0$ .

На языке C/C++ окончательно логическое выражение будет иметь вид:

`x >= 0 && y >= 0 && x*x+y*y <= 4 || x >= (-2) && x <= 0 && y >= (-2) && y <= 0`

Схема алгоритма решения задачи представлена на рис. 2.1.

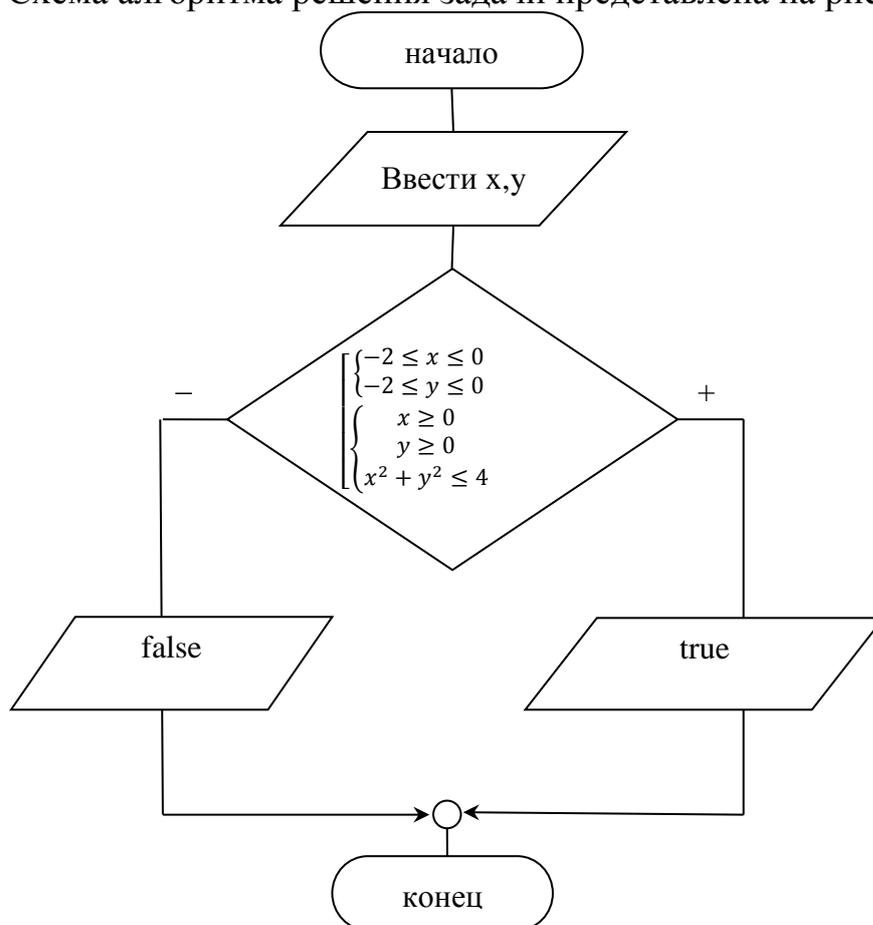


Рис. 2.1 Блок-схема решения задания 2.2

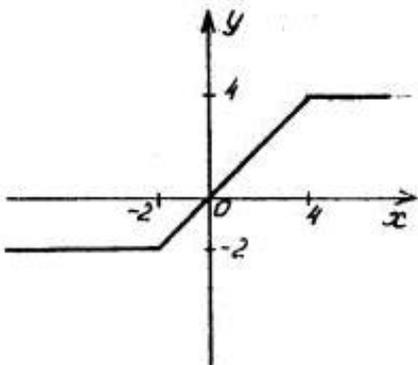
Текст программы:

```
#include <iostream>
using namespace std;
int main(){
double x,y;
cout<<"Enter A(x,y):\n";
cin>>x>>y;
bool s1,s2;
s1=x>=0 && y>=0 && x*x+y*y<=4 ;
s2=x>=(-2) && x<=0 && y>=(-2) && y<=0;
if (s1 || s2)
    cout<<"true";
    else cout<<"false";
return 0;
}
```

### Результаты работы программы:

```
Enter A(x,y) :
1
1
true
```

*Задание 2.2* Составить программу вычисления значения функции, заданной графически. (значение аргумента  $X$  вводится произвольно с клавиатуры).



Математическая модель:

$$y = \begin{cases} -2, & \text{при } x \leq -2; \\ x, & \text{при } -2 < x \leq 4; \\ 4, & \text{при } x > 4. \end{cases}$$

Схема алгоритма решения задачи представлена на рис.2.2.

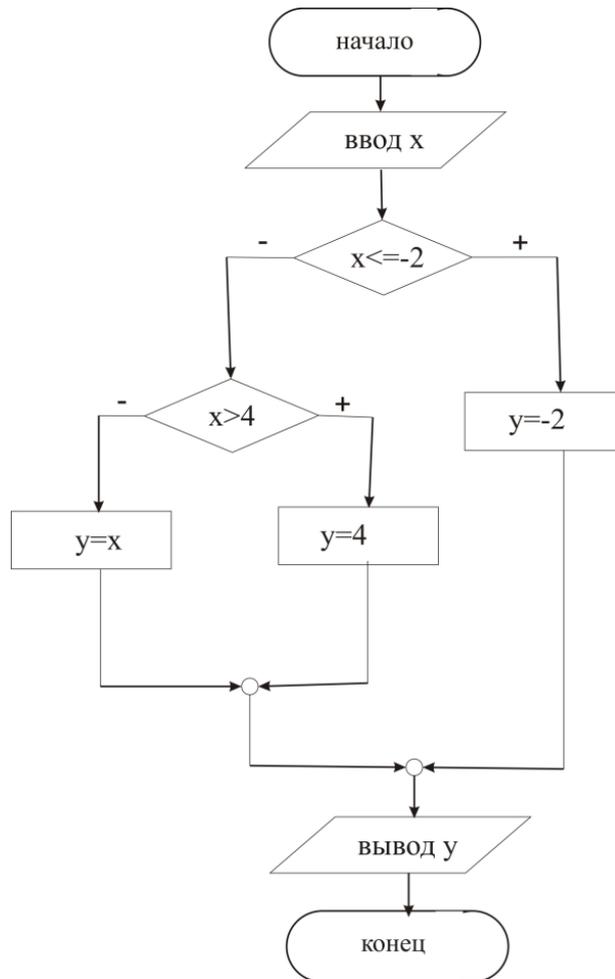


Рис. 2.2 Блок-схема решения задания 2.3

Текст программы:

```

#include <iostream>
using namespace std;
int main() {
double x, y;
cout << "Enter x: \n";
cin >> x;
if (x <= -2) y = -2;
    else if (x > 4) y = 4;
        else y = x;
    cout << "y=" << y;
    return 0;
}
  
```

Результаты работы программы:

```

Enter x:
6
y=4
  
```