

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Рязанский государственный радиотехнический университет
имени В. Ф. Уткина»

Кафедра «Вычислительная и прикладная математика»

Отчет
по лабораторной работе № 4
по дисциплине
«Низкоуровневое программирование»
на тему
«Указатели»

Выполнил:
студент гр. 143
Вербицкая И. С.

Проверил:
Щенева Ю.Б.

Рязань 2022

Задание (вариант №5):

Задание

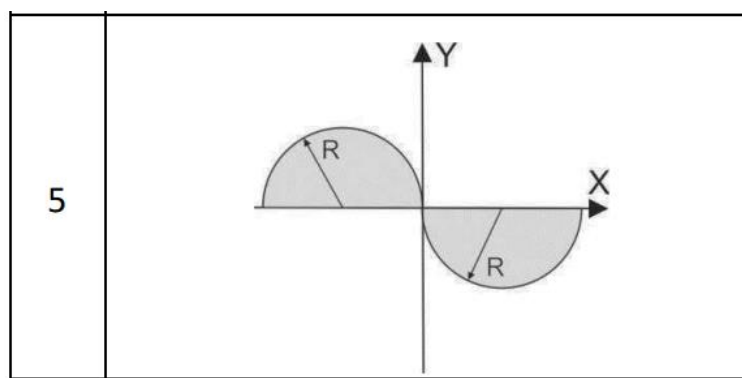
Преобразовать лабораторную работу №3 таким образом, чтобы работа с массивами велась только через арифметику указателей. Выбор функции заполнения массива необходимо осуществлять через указатель на функцию.

Дан массив координат, необходимо определить какие из этих координат попадают в заданную закрашенную область, а какие не попадают, и вывести эту информацию на экран. Массив координат задаётся двумерным массивом, где первый индекс означает номер координаты, а второй номер компоненты координаты ($x - 0$, $y - 1$). Массив задаётся по выбору пользователя либо через непосредственный ввод координат с клавиатуры, либо случайным образом в заданном пользователем диапазоне.

В программе должно быть минимум четыре отдельных функции:

1. Задание массива координат через непосредственный ввод координат с клавиатуры.
2. Задание массива координат случайным образом в заданном диапазоне.
3. Определение попадания точки в заданную область.
4. Вывод попавших и не попавших координат в область.

Параметры R , a и b , задающие область, а также количество координат задаются пользователем.



Анализ задания:

Для попадания точки в изображённую область должно выполняться 2 условия:

1. Попадание точки в одну из окружностей радиуса R с центрами $(0;R)$ и $(0;-R)$.
Это условие задается неравенством $(x \pm R)^2 + y^2 \leq R^2$
2. Попадание точки во II или IV четверти координатной плоскости.
Это условие задается неравенством $xy \leq 0$

В результате для изображённой области получаем систему неравенств:

$$\begin{cases} (x \pm R)^2 + y^2 \leq R^2 \\ xy \leq 0 \end{cases}$$

Этапы решения задачи:

1. Составить блок-схему;
2. Составить программу;
3. Провести проверку работы программы.

Блок-схема:

На рисунке 1 изображена блок-схема подпрограммы заполнения массива координат случайными числами, на рисунке 2 – заполнения вручную.

На рисунке 3 изображена блок-схема функции, возвращающей 0 в случае, когда заданная точка не попадает в область, и 1 – когда попадает.

На рисунке 4 изображена блок-схема подпрограммы вывода попавших и не попавших в область координат.

На рисунке 5 изображена блок-схема основной программы.

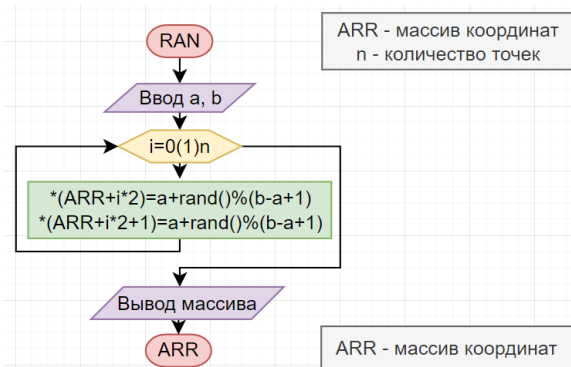


Рисунок 1

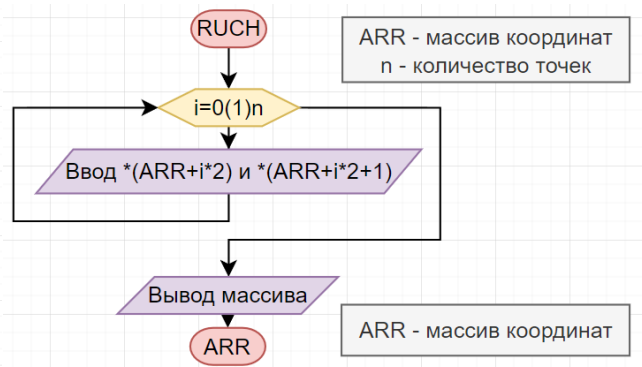


Рисунок 2

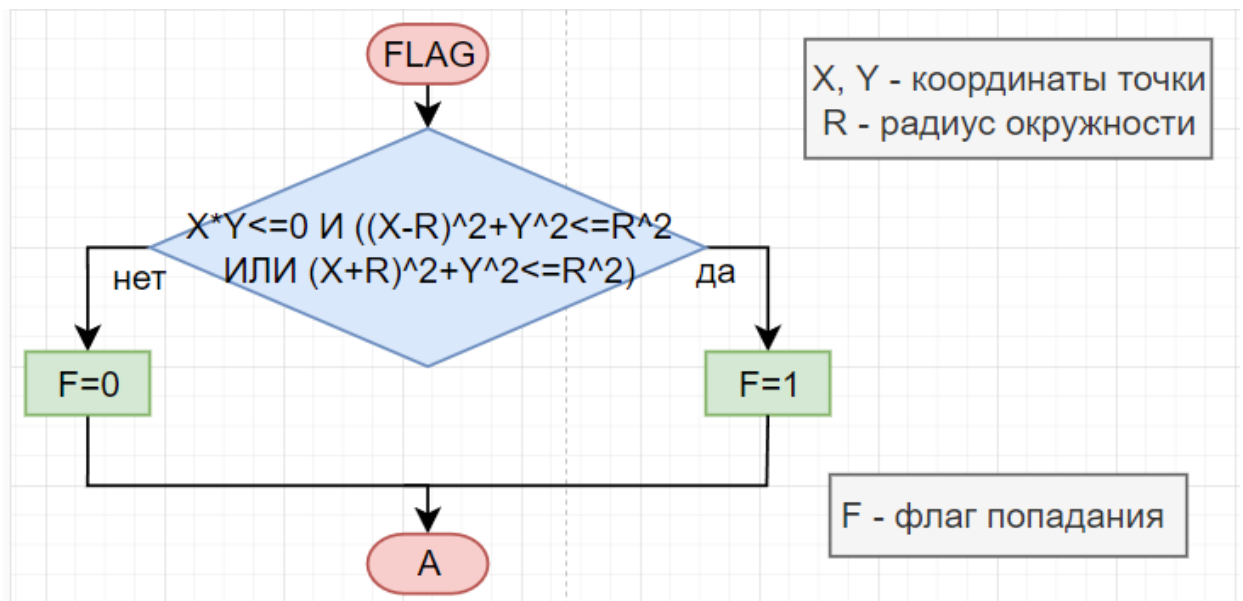


Рисунок 3

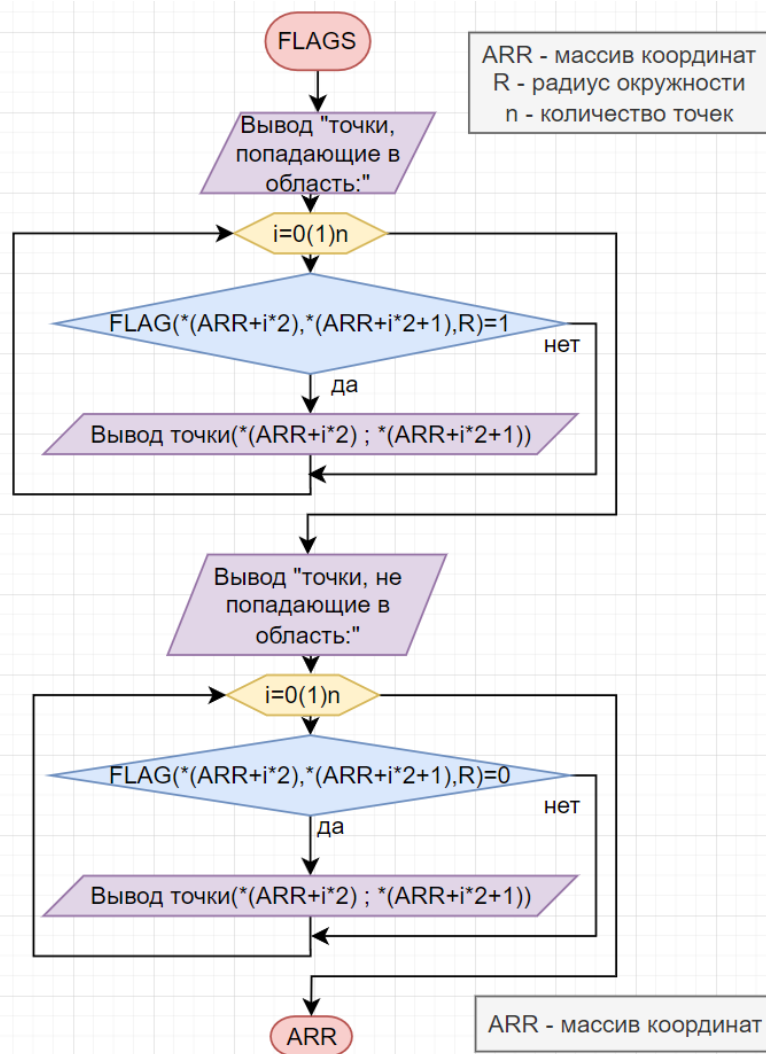


Рисунок 4

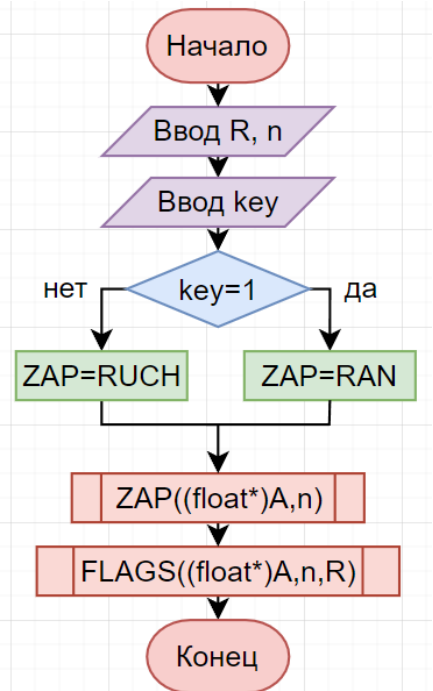


Рисунок 5

Листинг программы (рисунок 6, рисунок 7):

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include <time.h>
5  /* x y
6   x y
7  */
8
9  //ФУНКЦИЯ СЛУЧАЙНОГО ЗАПОЛНЕНИЯ ЗНАЧЕНИЙ КООРДИНАТ
10 void RAN (float* ARR, int n)
11 {
12     int a,b;
13     printf("Введите нижнюю границу диапазона случайных чисел, a=");
14     scanf("%d",&a);
15     printf("Введите верхнюю границу диапазона случайных чисел, b=");
16     do {
17         scanf("%d",&b);
18         if (b<=a)
19             printf("Ошибка: значение конца отрезка должно быть больше значения начала, b=");
20     } while (b<=a);
21     srand(time(NULL));
22     int i;
23     for (i=0;i<n;i++)
24     {
25         *(ARR+i*2)=a+rand()%(b-a+1);
26         *(ARR+i*2+1)=a+rand()%(b-a+1);
27     }
28     printf("\nИСХОДНЫЕ ЗНАЧЕНИЯ:\n");
29     printf("      X      Y\n");
30     for(i=0;i<n;i++)
31         printf("%.1f %.1f\n",7,*(ARR+i*2),7,*(ARR+i*2+1));
32 }
33
  
```

Рисунок 6

```

33
34 //ФУНКЦИЯ РУЧНОГО ЗАПОЛНЕНИЯ ЗНАЧЕНИЙ X И Y
35 void RUCH (float* ARR, int n)
36 {
37     int i;
38     for (i=0;i<n;i++)
39     {
40         printf("Координаты точки №%d: ", i+1);
41         scanf("%f %f",&*(ARR+i*2),&*(ARR+i*2+1));
42     }
43     printf("\nИСХОДНЫЕ ЗНАЧЕНИЯ:\n");
44     printf("      X      Y\n");
45     for(i=0;i<n;i++)
46     printf("%.1f %.1f\n",*(ARR+i*2),*(ARR+i*2+1));
47 }
48
49 //ФУНКЦИЯ ОПРЕДЕЛЕНИЯ ПОПАДАНИЯ КОНКРЕТНОЙ ТОЧКИ В ЗАДАННУЮ ОБЛАСТЬ
50 float FLAG(float X, float Y, float R)
51 {
52     float F;
53     if ((X*Y<=0) && ((powf(X-R,2)+Y*Y<=R^2) || (powf(X+R,2)+Y*Y<=R^2)))
54         F=1;
55     else F=0;
56     return F;
57 }
58
59 //ФУНКЦИЯ ВЫВОДА ПОПАВШИХ И НЕ ПОПАВШИХ В ОБЛАСТЬ КООРДИНАТ
60 void FLAGS(float* ARR, int n, float R)
61 {
62     printf("\nТОЧКИ, ПОПАДАЮЩИЕ В ОБЛАСТЬ:\n");
63     int i;
64     for(i=0;i<n;i++)
65     if (FLAG(*(ARR+i*2),*(ARR+i*2+1),R)==1)
66     printf("%.1f ; %.1f\n",*(ARR+i*2),*(ARR+i*2+1));
67     printf("ТОЧКИ, НЕ ПОПАДАЮЩИЕ В ОБЛАСТЬ:\n");
68     for(i=0;i<n;i++)
69     if (FLAG(*(ARR+i*2),*(ARR+i*2+1),R)==0)
70     printf("%.1f ; %.1f\n",*(ARR+i*2),*(ARR+i*2+1));
71 }
72
73
74 //ОСНОВНАЯ ПРОГРАММА
75 int main()
76 {
77     system("chcp 1251");
78     float A [1000][2];
79     //задаем значение R
80     printf("Введите радиус окружностей, R=");
81     float R;
82     do {
83         scanf("%f",&R);
84         if (R<=0) |
85             printf("Ошибка: радиус должен быть больше нуля, R=");
86     } while (R<=0);
87     //ввод количества координат
88     printf("Введите количество точек, n=");
89     int n;
90     do {
91         scanf("%d",&n);
92         if (n<=0)
93             printf("Ошибка: количество должно быть больше нуля, n=");
94         if (n>1000)
95             printf("Ошибка: количество не может быть больше тысячи, n=");
96     } while ((n>1000) || (n<=0));
97     //выбор и реализация способа заполнения массива
98     int key;
99     void (*ZAP)(float*, int);
100     printf("Введите 1 - для случайных значений x и y;\nлюбое другое число - ручной ввод значений x и y:\n");
101     scanf("%d",&key);
102     if (key==1) ZAP=RAN; else ZAP=RUCH;
103     ZAP((float*)A,n);
104
105     //выводим попавшие и не попавшие точки
106     FLAGS((float*)A,n,R);
107     return 0;
108 }

```

Рисунок 7

Результаты работы программы и проверка:

Программа была запущена для значений $R=10$, $n=5$, $x_1=1$, $y_1=1$, $x_2=0$, $y_2=0$, $x_3=-5$, $y_3=5$, $x_4=5$, $y_4=-5$, $x_5=5$, $y_5=5$ (рисунок 8). Результаты, полученные с помощью программы, совпали с результатами, полученными с помощью Excel (рисунок 9).

```
Текущая кодовая страница: 1251
Введите радиус окружностей, R=10
Введите количество точек, n=5
Введите 1 - для случайных значений x и y;
любое другое число - ручной ввод значений x и y:
2
Координаты точки №1: 1 1
Координаты точки №2: 0 0
Координаты точки №3: -5 5
Координаты точки №4: 5 -5
Координаты точки №5: 5 5

ИСХОДНЫЕ ЗНАЧЕНИЯ:
  X      Y
  1.0    1.0
  0.0    0.0
 -5.0    5.0
  5.0   -5.0
  5.0    5.0

Точки, попадающие в область:
(0.0 ; 0.0)
(-5.0 ; 5.0)
(5.0 ; -5.0)
Точки, НЕ попадающие в область:
(1.0 ; 1.0)
(5.0 ; 5.0)

-----
Process exited after 52.26 seconds with return value 0
Для продолжения нажмите любую клавишу . . .
```

Рисунок 8

	A	B	C	D
1	R	x	y	Попадание
2	10	1	1	нет
3		0	0	да
4		-5	5	да
5		5	-5	да
6		5	5	нет

Рисунок 9