

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа № 3
«Одномерные массивы»

Проверил:
Богдан Е. В.

Выполнил:
Горох А.А.

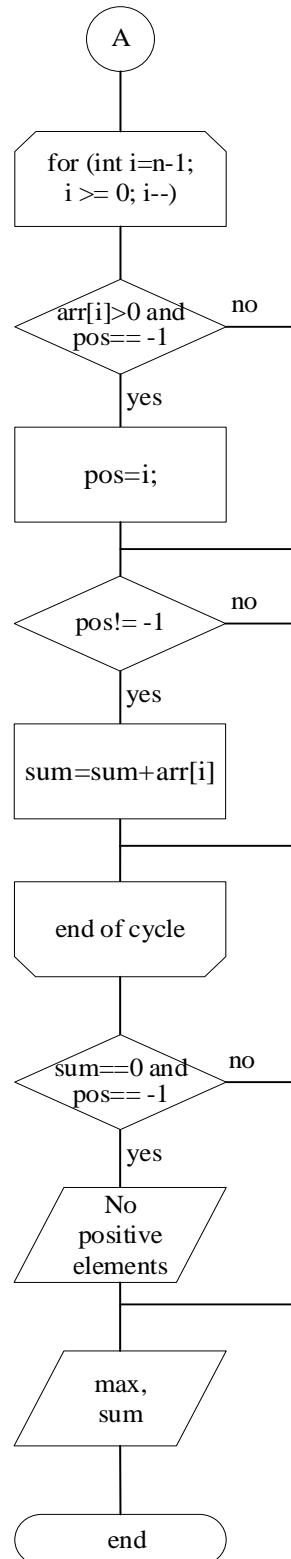
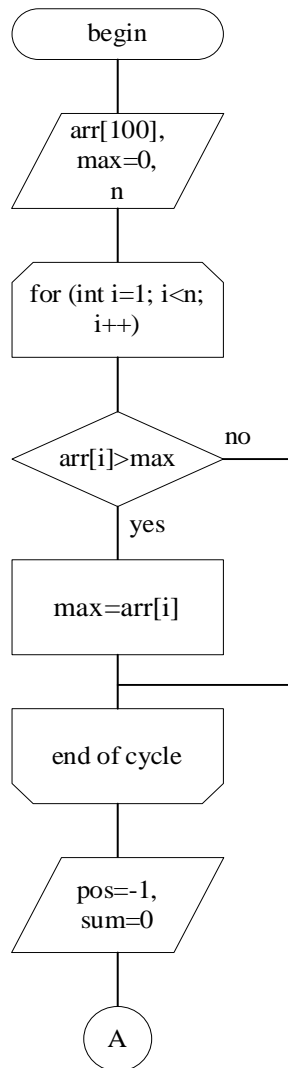
МИНСК 2022

Ход работы

1. **Задача 1.3.** В одномерном массиве, состоящем из n вещественных элементов, вычислить:

- максимальный элемент массива;
- сумму элементов массива, расположенных до последнего положительного элемента.

Блок-схема:



Исходный код:

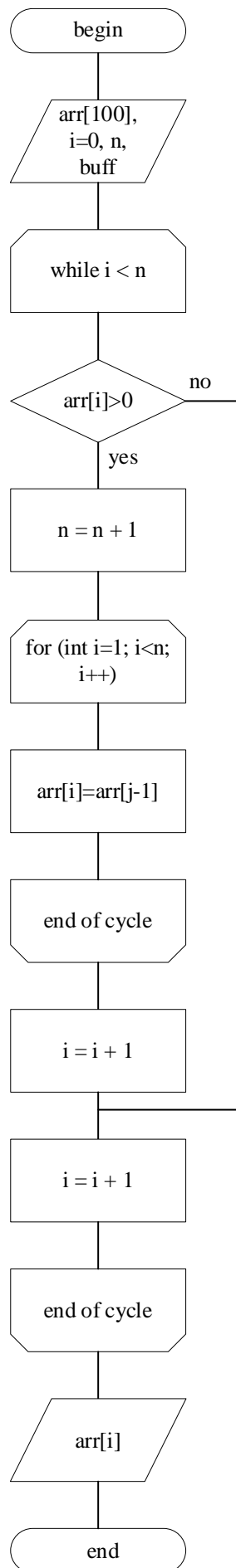
```
void task_1()
{
    float arr[100];
    int n;
    printf("Enter the number of array elements: ");
    n=pos_int();
    init_f(arr,n);
    printf("\n");
    float max=0;
    for(int i=0;i<n;i++)
    {
        if(arr[i]>max)
            max=arr[i];
    }
    printf("Maximum member of array is %.2f\n",max);
    int pos=-1;
    float sum=0;
    for(int i=n-1;i>=0;i--)
    {
        if (arr[i]>0 && pos== -1)
            pos=i;
        if(pos!= -1)
            sum+=arr[i];
    }
    if(sum==0 && pos== -1)
        printf("Sorry, there is no such element\n");
    printf("Amount of members up to last positive element:
%.2f\n", sum);
}
```

Результат выполнения программы:

```
Choose number of task(enter -1 for exit): 1
Enter the number of array elements: 5
Choose way of array filling(1-for user input, 2-for random filling): 2
-76.27 17.46 -46.76 130.37 197.09
Maximum member of array is 197.09
Amount of members up to last positive element: 221.89
```

Задача 2. 3. В массиве из n элементов после каждого отрицательного элемента добавить его копию.

Блок-схема:



Исходный код:

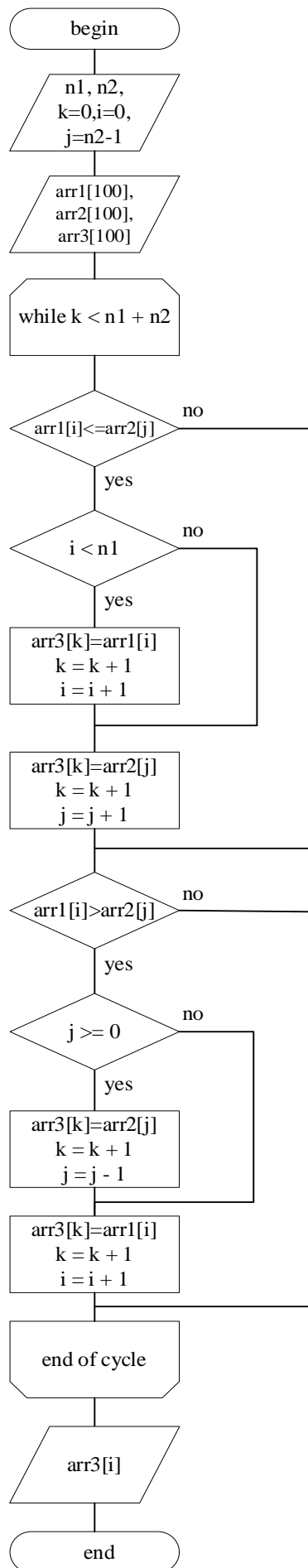
```
void task_2()
{
    float arr[100];
    int n;
    printf("Enter the number of array elements: ");
    n=pos_int();
    init_f(arr,n);
    printf("\n");
    int i=0;
    float buff;
    while(i<n)
    {
        if(arr[i]<0)
        {
            n++;
            for(int j=n-1;j>i;j--)
            {
                arr[j]=arr[j-1];
            }
            i++;
        }
        i++;
    }
    for(int i=0;i<n;i++)
    {
        printf("%.2f ",arr[i]);
    }
}
```

Результат выполнения программы:

```
Choose number of task(enter -1 for exit): 2
Enter the number of array elements: 6
Choose way of array filling(1-for user input, 2-for random filling): 2
-71.50 -18.51 148.21 201.05 183.09 151.15
-71.50 -71.50 -18.51 -18.51 148.21 201.05 183.09 151.15
```

Задача 3. 3. Имеется два массива целых чисел: первый заполнен по возрастанию, второй - по убыванию. Объединить массивы в третий массив в порядке возрастания.

Блок-схема:



Исходный код:

```
void task_3() {
int arr1[100],arr2[100],arr3[200];
int n1, n2, diff;
diff=1;
printf("Enter the number of first array elements: ");
n1=pos_int();
init_int(arr1,n1,diff);
diff=-1;
printf("Enter the number of second array elements: ");
n2=pos_int();
init_int(arr2,n2,diff);
printf("\n");
int k=0,i=0,j=n2-1;
while(k<n1+n2) {
    if(arr1[i]<=arr2[j])
    {
        if(i<n1)
        {
            arr3[k]=arr1[i];
            k++;
            i++;
        }
        else {
            arr3[k]=arr2[j];
            k++;
            j++;
            continue;
        }
    }
    if(arr1[i]>arr2[j])
    {
        if(j>=0)
        {
            arr3[k]=arr2[j];
            k++;
            j--;
        }
        else
        {
            arr3[k]=arr1[i];
            k++;
            i++;
            continue;
        }
    }
}
for(int i=0;i<n1+n2;i++)
    printf("%d ",arr3[i]);
}
```

Результат выполнения программы:

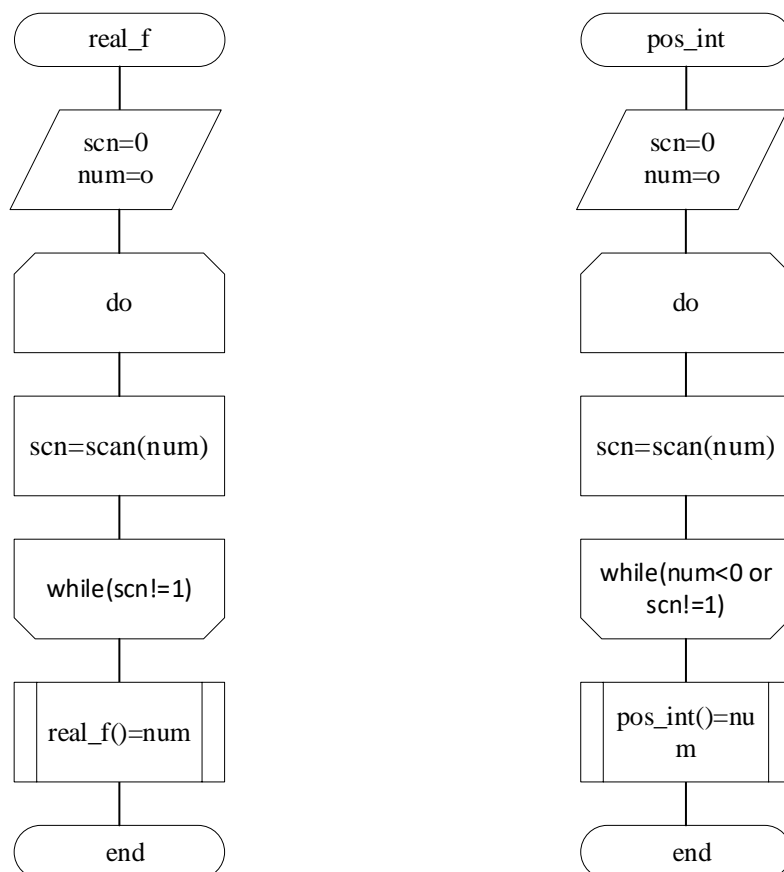
```

Choose number of task(enter -1 for exit): 3
Enter the number of first array elements: 7
Choose way of array filling(1-for user input, 2-for random filling): 2
5 95 190 236 295 371 450
Enter the number of second array elements: 6
Choose way of array filling(1-for user input, 2-for random filling): 2
21 -43 -55 -114 -210 -251
The third array:
-251 -210 -114 -55 -43 5 21 95 190 236 295 371 450

```

Весь ввод размерности массива и самих массивов проводился через функции `real_f` и `pos_int`.

Блок-схема:



Исходный код:

```

float real_lf() {
    int scn=0;
    double num=0;
    do {
        scn=scanf("%lf", &num);
        while (getchar() != '\n');
        if(scn!=1)
            printf("Wrong input\n");
    }
}

```



```

    }
    while (scn!=1);
    return num;
}
int pos_int() {
    int scn=0;
    int num=0;
    do {
        scn=scanf("%d",&num);
        while (getchar() != '\n');
        if (scn!=1 || num<=0)
            printf("Wrong input\n");
    }
    while (num<=0 || scn!=1);
    return num;
}

```

Для упрощенной проверки все задачи собраны в один файл через функции `void task"номер задачи"()` и каждая задача вызывается через оператор вывода `switch ... case`.

Исходный код:

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
int main() {
    int task=0;
    while (task!=-1) {
        printf("Choose number of task(enter -1 for exit): ");
        scanf("%d",&task);{
            switch (task)
            {
                case 1: {
                    task_1();
                    break;
                }
                case 2: {
                    task_2();
                    break;
                }
                case 3: {
                    task_3();
                    break;
                }
            }
        }
    }
}

```

```
    }  
    case -1: {  
        exit(0);  
    }  
    default:  
        printf("Incorrect input number of task, try again");  
        }  
        getch();  
        system("cls");  
    }  
}  
getch();  
return 0;  
}
```