

Лабораторная работа №9.  
ИЦВП с управлением по индексу и функции.

Цель: средствами языка C научиться работать с ИЦВП по индексу и функции.

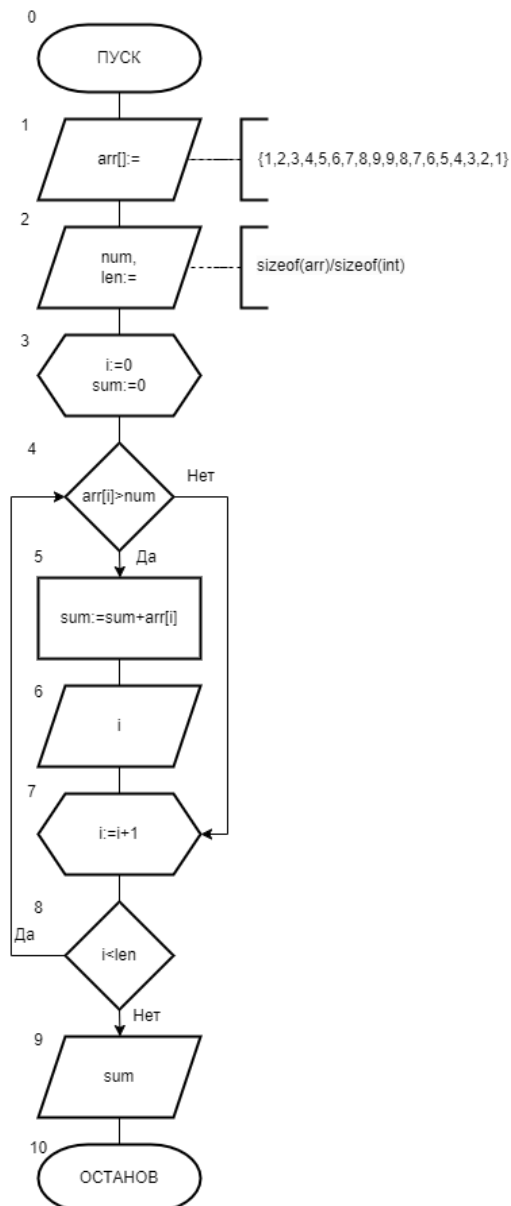
Используемое оборудование: ПК, Visual Studio Code

Задача 1: Дан одномерный массив. С клавиатуры вводится число. Найти сумму всех элементов массива, значения которых больше числа, введенного с клавиатуры. Вывести итоговую сумму и индексы этих элементов.

Математическая модель:

$$S = \sum_{i=1}^n a_i \cdot \delta(a_i > x)$$
$$I = \{i \mid a_i > x, i = 1, 2, \dots, n\}$$

Блок схема:



Список идентификаторов:

Имя	Тип	Смысл
Arr[]	Int	Массив
Num	Int	Вводимое число
Len	Int	Длинна массива
Sum	Int	Сумма элементов, подходящих под условие
I	Int	Параметр цикла

Код программы:

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void){
5
6  int arr[] = {1,2,3,4,5,6,7,8,9,9,8,7,6,5,4,3,2,1};
7  int num,sum;
8  sum = 0;
9  int len = sizeof(arr)/sizeof(int);
10 printf("Enter number: ");
11 scanf("%d", &num);
12 printf("Indexes: ");
13 for(int i = 0; i < len; i++){
14     if (arr[i] > num){
15         sum+=arr[i];
16         printf(" %d",i);
17     }
18 }
19 printf("\nSum: %d",sum);
20 }
```

Результат работы программы:

```
Enter number: 5
Indexes:  5 6 7 8 9 10 11 12
Sum: 60
```

Анализ вычислений:

Был создан произвольный массив, переменная для суммы. Вводится число, после чего каждый элемент сравнивается с ним по очереди, при выполнении условия выводится индекс элемента, и он добавляется в сумму. В конце выводиться сумма.

Вывод:

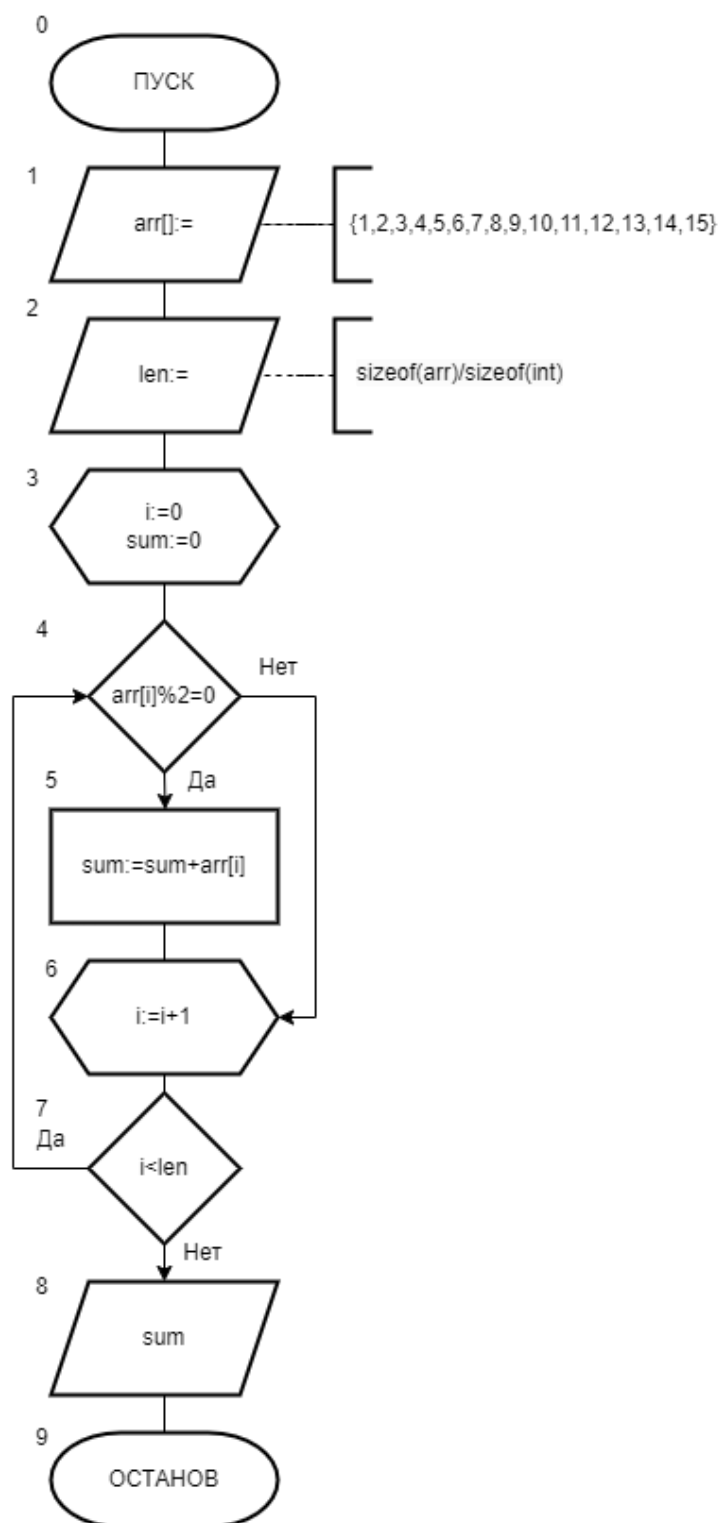
средствами языка C я научился работать с ИЦВП по индексу и функции.

Задача 2: Дан одномерный массив. Найти сумму четных по значению элементов массива.

Математическая модель:

$$S = \sum_{a_i \text{ четное}} a_i$$

Блок схема:



Список идентификаторов:

Имя	Тип	Смысл
Arr[]	Int	Массив
Len	Int	Длина массива
Sum	Int	Сумма четных элементов массива
I	Int	Параметр цикла

Код программы:

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void){
5
6  int arr[] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
7  int sum = 0;
8  int len = sizeof(arr)/sizeof(int);
9  for (int i = 0; i < len; i++)
10 {
11     if(arr[i]%2==0){
12         sum += arr[i];
13     }
14 }
15 printf("Sum: %d",sum);
16 }
```

Результат работы программы:

```
Sum: 56
```

Анализ вычислений:

Был создан произвольный массив, переменная для суммы. Каждый элемент массива проверяется на четность, и в случае прохождения проверки складывается с переменной суммы. Выводится результат.

Вывод:

средствами языка С я научился работать с ИЦВП по индексу и функции.

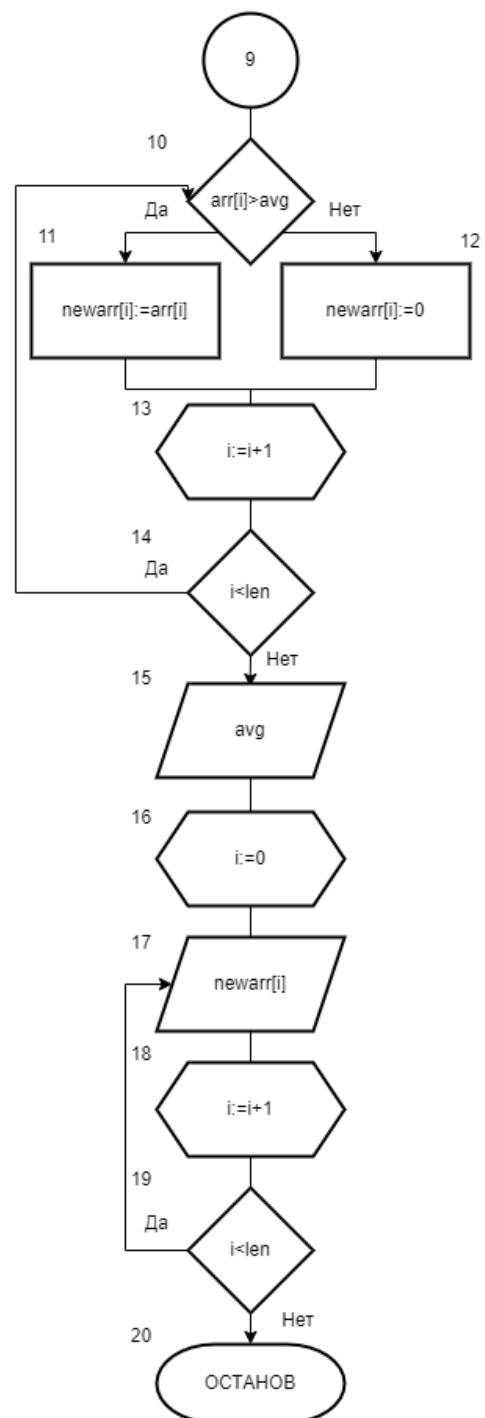
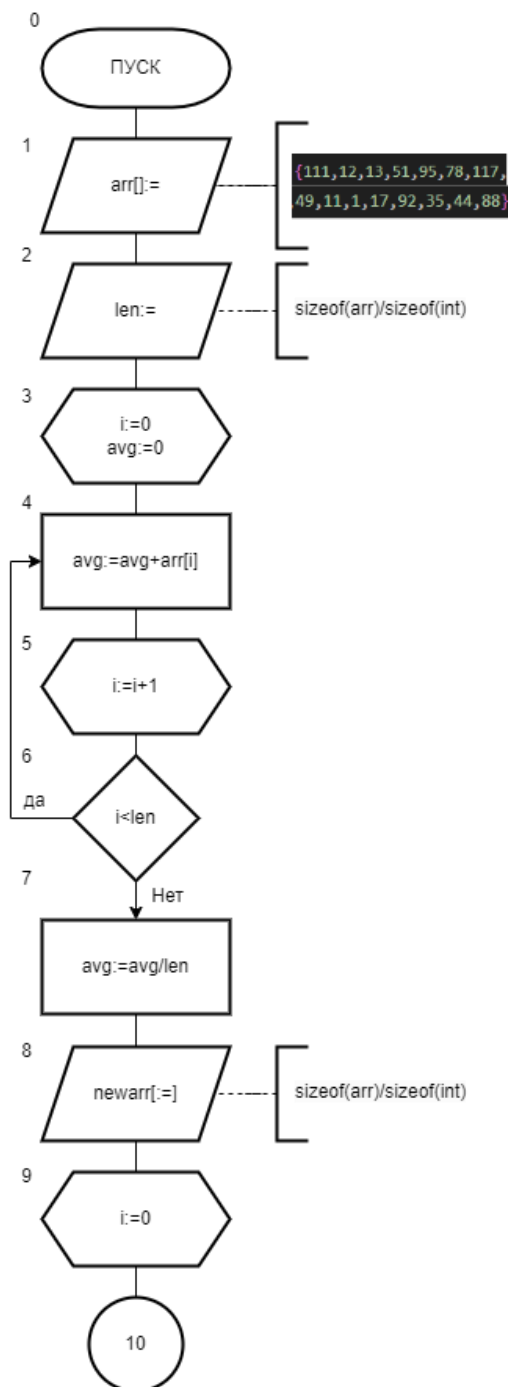
Задача 3: Дан одномерный массив. Найти его среднее арифметическое. Составить второй массив, элементами которого будут элементы первого массива, которые больше среднего арифметического. Остальные элементы заменить нулями.

Математическая модель:

$$\text{avg} = \frac{\sum a_i}{n}$$

$$\text{newarr}_i = \begin{cases} a_i, & a_i > \text{avg} \\ 0, & a_i \leq \text{avg} \end{cases}$$

Блок схема:



Список идентификаторов:

Имя	Тип	Смысл
Arr[]	Double	Первичный массив
Newarr[]	Double	Вторичный массив
Avg	Double	Среднее арифметическое элементов массива
Len	Double	Длина массива
I	Double	Параметр цикла

Код программы:

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main(void){
5
6      int arr[] = {111,12,13,51,95,78,117,49,11,1,17,92,35,44,88};
7      double len = sizeof(arr)/sizeof(int);
8      double avg = 0;
9
10     for (int i = 0; i < len; i++)
11     {
12         avg += arr[i];
13     }
14     avg /= len;
15
16     int newarr[sizeof(arr)/sizeof(int)];
17     for (int i = 0; i < len; i++)
18     {
19         if (arr[i]>avg){
20             newarr[i] = arr[i];
21         }
22         else{
23             newarr[i]=0;
24         }
25     }
26
27     printf("Avg: %lf",avg);
28     printf("\nnewarr:");
29     for (int i = 0; i < len; i++)
30     {
31         printf(" %d",newarr[i]);
32     }
33
34 }
```

Результат работы программы:

```
Avg: 54.266667
newarr: 111 0 0 0 95 78 117 0 0 0 0 92 0 0 88
```

Анализ вычислений:

Был создан произвольный массив. Считается среднее арифметическое его элементов. Создается новый массив такого-же размера. Цикл идет по длине массива, если элемент подходит, он присваивается в новый массив, если нет, то присваивается ноль. Выводится итоговый массив.

Вывод:

средствами языка С я научился работать с ИЦВП по индексу и функции.