**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Вычислительной техники**

отчет

**по лабораторной работе № 2**

**по дисциплине «Программирование»**

Тема: **Обработка одномерных массивов**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3312 |  | Мохно Даниил. |
| Преподаватель |  | Аббас Саддам |

Санкт-Петербург

2023

**Цель работы.**

Целью работы является изучение **обработки одномерных массивов на языке Си** и получение практических навыков в программировании на этом языке.

**Задание (вариант 1)**

Ввести массив чисел, количество которых заранее неизвестно. Ввод массива заканчивается вводом числа, имеющего противоположный знак по сравнению с первым введенным числом. Без использования другого массива переставить элементы в первой половине массиве в противоположном порядке, а в оставшейся части удалить элементы, значение которых четно. В случае нечетного количества элементов в массиве средний элемент остается на своем месте. Вывести преобразованный массив.

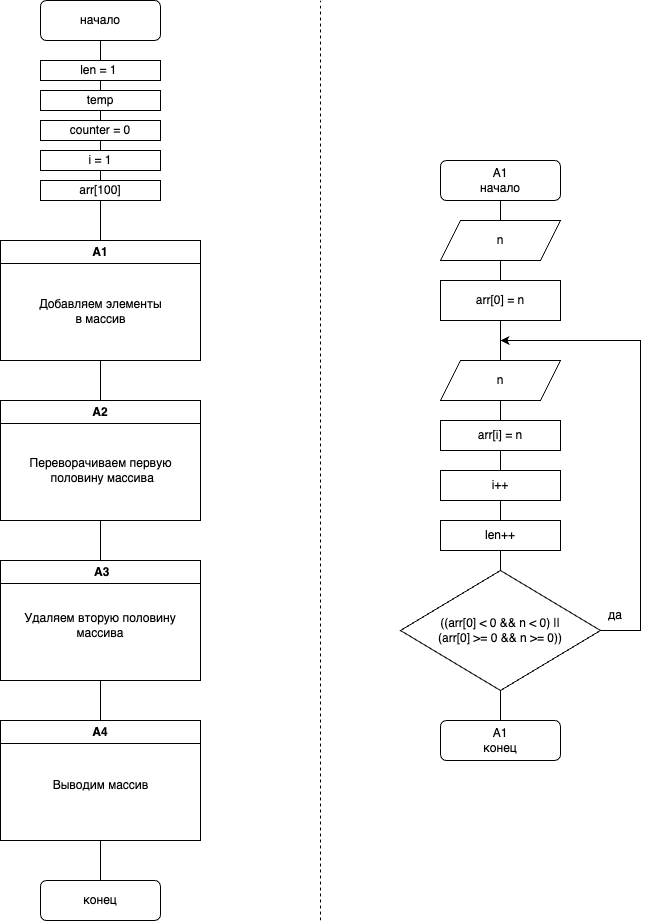
**Постановка задачи и описание решения**

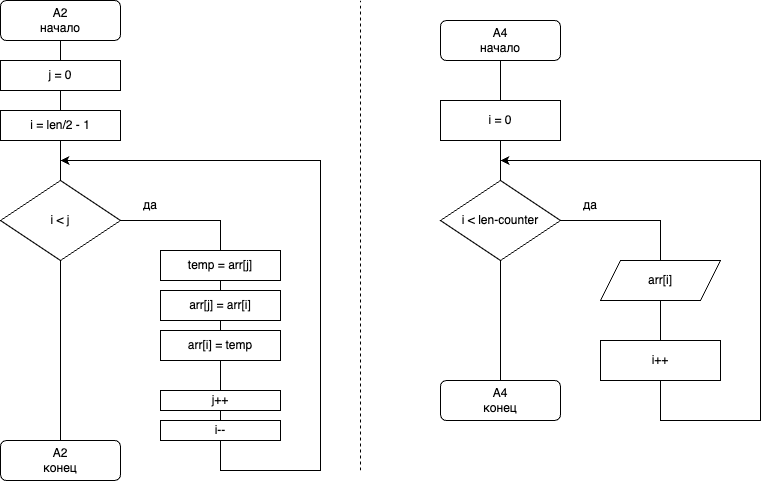
Для начала необходимо создать целочисленный массив и ввести его нулевой элемент. Через цикл с пост условием мы будем заполнять массив проверяя чтобы и нулевой элементы и введённое число были оба либо больше нуля либо меньше нуля (имеет ли введённое число противоположный нулевому элементу знак). Далее переворачиваем первую половину массива. Для этого итерируем массив, беря первый индекс с начала массива и инкрементируя его с каждой итерацией, а второй индекс с пред-серединного элемента (т.к. центральный элемент, в случае, когда массив имеет чётное количество, остаётся на месте) декрементируя его с каждой итерацией. Меняем элементы местами, временно сохраняя в дополнительную переменную значение по первому индексу и переставляя на его место значение по второму индексу, а на место значения по второму индексу ставим значение из дополнительной переменной. Для удаления чётных элементов второй половины массива нужно переставить их в конец массива и вывести полученный массив без них. Для перемещения чётных элементов в конец, используем вложенный цикл. В случае когда у нас чётное количество элементов в массиве, первый итератор идёт начиная со второй половины массива, т. е. в начале он будет равен длине массива делённой на 2, в случае когда у нас нечётное количество элементов в массиве, первый итератор будет начинаться с длинны массива делённой на 2 + 1, так как средний элемент массива должен остаться не тронутым, второй итератор начинается с первого итератора. Далее проверяем, чтобы остаток от деления числа на 2 был равен 0 (делится ли число на 2), и делаем перестановку элементов, с сохранением в промежуточную переменную, аналогично тому, как мы сделали в предыдущем цикле. Далее пройдёмся циклом по второй половине массива в убывающем порядке, чтобы сосчитать количество чётных элементов, находящихся в конце второй половины массива, увеличивая переменную счётчик, когда встречаем число, дающее 0 в остатке при делении на 2. В случае, когда в массиве чётное количество элементов, итератор будет начинаться с длинны массива, а когда количество элементов не чётное, итератор начинается с длинны массива – 1, чтобы счётчик не доходил до центрального элемента. И наконец выводим массив, вычтя из его длинны количество чётных элементов второй половины.

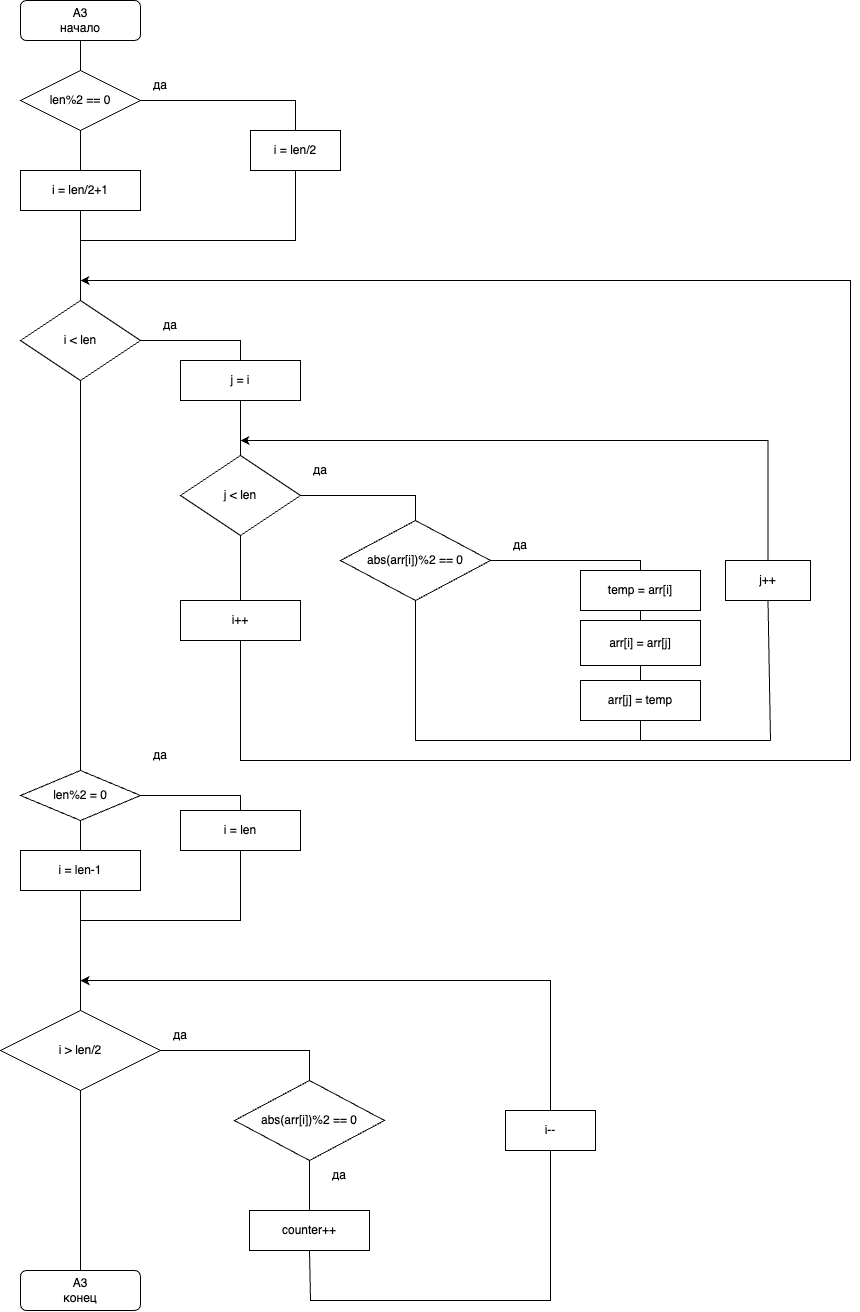
**Описание переменных**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя переменно | Тип | Назначение |
| 1 | len | int | Длинна будущего массива. |
| 2 | n | int | Входное число |
| 3 | temp | int | Переменная для временного хранения переменной из массива, во время сортировки |
| 4 | i | int | Итератор |
| 5 | j | int | Итератор |
| 6 | arr[100] | int | Массив, собственной персоной |
| 7 | counter | int | счётчик чётных элементов второй половины массива |

**Схема алгоритма**

****

****

****

**Контрольные примеры**

|  |  |  |
| --- | --- | --- |
| Пример № | Входные данные | Выходные данные |
| 1 | 89 43 94 23 11 42 90 87 65 56 96 39 44 -9 | 90 42 11 23 94 43 89 87 65 39 -9 |
| 2 | -90 -89 -65 -88 -34 -9 -22 -55 -1 6 | -34 -88 -65 -89 -90 -9 -55 -1 |
| 3 | 5 45 66 7 89 44 45 221 13 42 86 9 -99 | 44 89 7 66 45 5 45 221 13 9 -99 |
| 4 | 8 -8 | 8 |

**Текст программы**

*#include* <stdio.h>  
*#include* <stdlib.h>  
  
int main()  
{  
 int len = 1;  
 int n;  
 int temp;  
 int counter = 0;  
 int i = 1;  
 int j;  
 int arr[100];  
  
 scanf("%i",&n);  
 arr[0] = n;  
  
 do  
 {  
 scanf("%i", &n);  
 arr[i] = n;  
 i++; len++;  
 } while ((arr[0] < 0 && n < 0) || (arr[0] >= 0 && n >= 0));  
  
 for(j=0, i=len/2-1; j<i; j++, i--)

{  
 temp = arr[j];  
 arr[j] = arr[i];  
 arr[i] = temp;  
 }  
  
 if (len%2 == 0)  
 i = len/2;  
 else  
 i = len/2+1;  
  
 for(; i<len; i++)

{  
 for (j = i; j<len; j++)

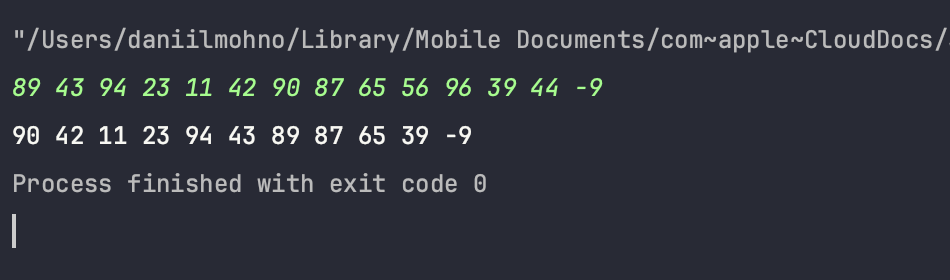
{  
 if (abs(arr[i])%2 == 0)  
 {  
 temp = arr[i];  
 arr[i] = arr[j];  
 arr[j] = temp;  
 }  
 }  
 }  
   
 if (len%2 == 0)

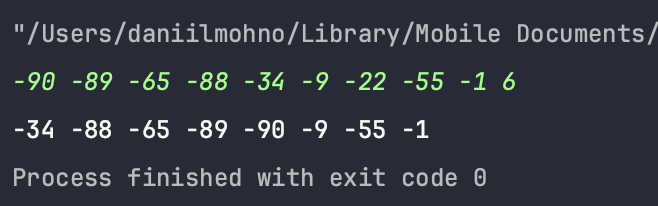
i = len;  
 else

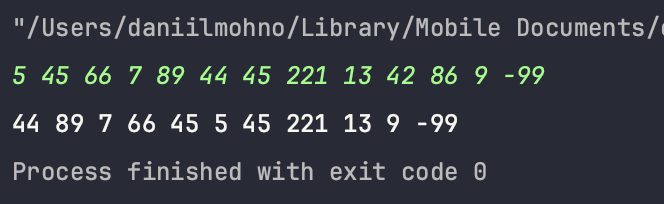
i = len-1;

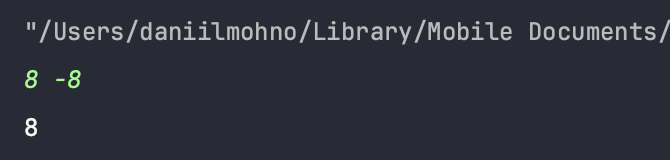
for (; i > len/2; i--)  
 if (abs(arr[i])%2 == 0) counter++;  
  
 for (i = 0; i < len-counter; i++)  
 printf("%i ", arr[i]);  
  
 return 0;  
}

**Примеры выполнения программы**

****

****

****

****

**Выводы.**

В результате выполнения работы была изучена **обработка одномерных массивов на языке Си** и получены практические навыки в программировании на этом языке.