|  |  |  |
| --- | --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования  Белорусский Государственный Университет Информатики и Радиоэлектроники | | |
| Факультет компьютерных систем и сетей | | |
| Кафедра электронных вычислительных средств | | |
| **ОТЧЕТ**по лабораторной работе №3 «Одномерные массивы.» | | |
| Выполнил  студ. гр.250702 (пг1)  Букато П.А. |  | Проверил  преп. каф. ЭВС  Скиба И.Г. |
| Минск 2022 | | |

# 1 Цель лабораторной работы

* 1. Цель работы – научиться разрабатывать алгоритмы для работы с одномерными массивами и писать код на языке Си по составленному алгоритму.

# 2 Результаты выполнения лабораторной работы

**Задание 1** В одномерном массиве, состоящем из n целых элементов, вычислить:  
- номер максимального элемента массива;  
- произведение элементов массива, расположенных между первым и вторым нулевыми элементами.

## 2.1 Результат выполнения задания 1

**2.1.1** На рисунке 2.1 приведена блок-схема алгоритма для выполнения задания №1.

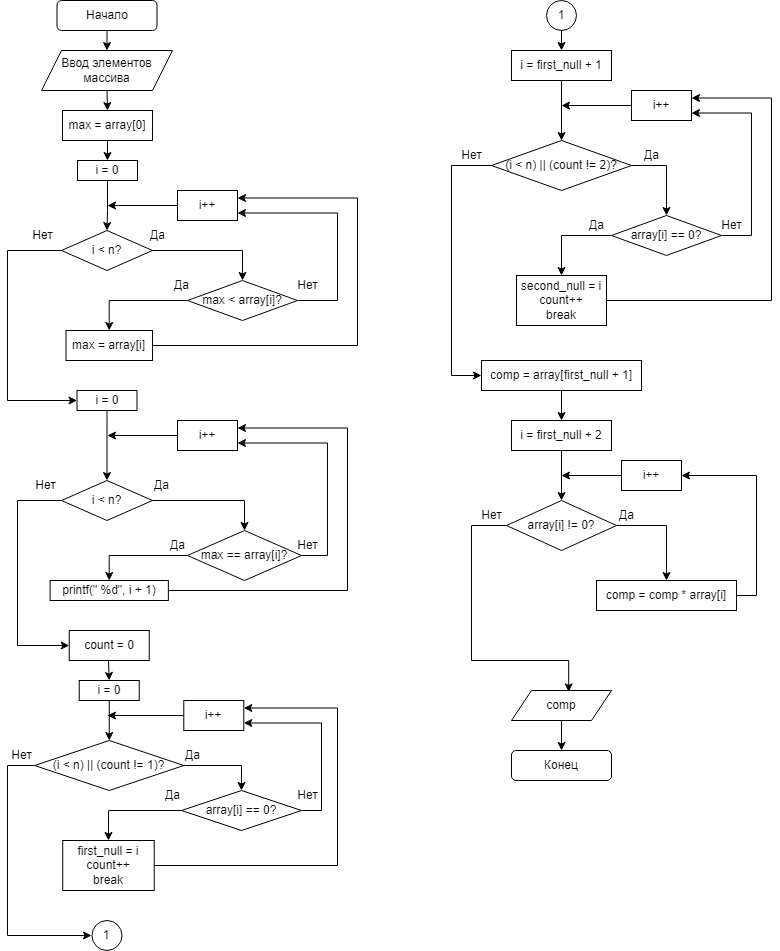


Рисунок 2.1 - Блок-схема алгоритма задания №1

**2.1.2** Листинг компьютерной программы по заданию 1.

#include <stdio.h>

int main()

{

int array[100], n, i, option;

printf("Input the length of the array\n");

scanf\_s("%d", &n);

printf("Choose an option:\n1) Keyboard input\n2) Initializing an array with random elements\n");

scanf\_s("%d", &option);

switch (option)

{

case 1:

printf("Input the elements of the array\n");

for (i = 0; i < n; i++)

{

scanf\_s("%d", &array[i]);

}

break;

case 2:

srand(time(NULL));

for (i = 0; i < n; i++)

{

array[i] = rand() % 30-15;

printf("%d ", array[i]);

}

printf("\n");

break;

default:

printf("Error");

return 0;

}

if (n <= 0)

{

printf("Invalid input");

return 0;

}

int max = array[0];

for (i = 0; i < n; i++)

{

if (max < array[i])

max = array[i];

}

printf("Numbers of max elements:");

for (i = 0; i < n; i++)

{

if (max == array[i])

{

printf(" %d", i + 1);

}

}

int first\_null, second\_null, count = 0;

for (i = 0; (i < n) || (count != 1); i++)

{

if (array[i] == 0)

{

first\_null = i;

count++;

break;

}

}

for (i = first\_null + 1; (i < n) || (count != 2); i++)

{

if (array[i] == 0)

{

second\_null = i;

count++;

break;

}

}

count = 0;

for (i = 0; (i < n); i++)

{

if (array[i] == 0)

count++;

}

if (count == 0)

{

printf("\nNo null elements in the array");

return 0;

}

if (count == 1)

{

printf("\nOnly one null element in the array");

return 0;

}

if (first\_null == second\_null - 1)

{

printf("\nNo elements between first and second null elements");

return 0;

}

int comp = array[first\_null + 1];

for (i = first\_null + 2; array[i] != 0; i++)

{

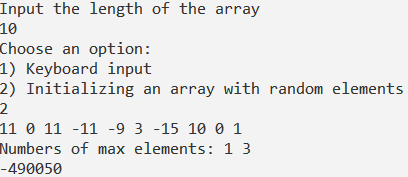
comp = comp \* array[i];

}

printf("\n%d", comp);

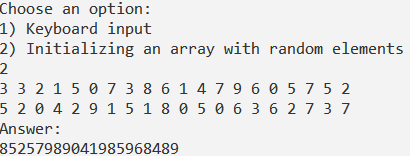
}

**2.1.3** Результат выполнения компьютерной программы в виде «скриншота» изображения на мониторе представлен на рисунке 2.2:



|  |  |  |
| --- | --- | --- |
|  |  |  |
| Рисунок 2.2 - Скриншот результата выполнения компьютерной программы | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Задание 2** В массиве из n элементов удалить элемент до каждого нечетного.  **2.2 Результат выполнения задания 2**  **2.2.1** На рисунке 2.3 приведена блок-схема алгоритма для выполнения задания №2.    Рисунок 2.3 - Блок-схема алгоритма задания №2  **2.2.2** Листинг компьютерной программы по заданию 2.  #include <stdio.h>  int main()  {  int array[100], n, i, option;  printf("Input the length of the array\n");  scanf\_s("%d", &n);  printf("Choose an option:\n1) Keyboard input\n2) Initializing an array with random elements\n");  scanf\_s("%d", &option);  switch (option)  {  case 1:  printf("Input the elements of the array\n");  for (i = 0; i < n; i++)  {  scanf\_s("%d", &array[i]);  }  break;  case 2:  srand(time(NULL));  for (i = 0; i < n; i++)  {  array[i] = rand() % 30-15;  printf("%d ", array[i]);  }  printf("\n");  break;  default:  printf("Error");  return 0;  }  if (n <= 0)  {  printf("Invalid input");  return 0;  }  for (i = 1; i < n + 1; i++)  {  if (array[i] % 2 == 0)  printf("%d ", array[i - 1]);  }  }  **2.2.3** Результат выполнения компьютерной программы в виде «скриншота» изображения на мониторе представлен на рисунке 2.4:     |  |  |  | | --- | --- | --- | |  |  |  | | Рисунок 2.4 - Скриншот результата выполнения компьютерной программы  **Задание 3** Для арифметических операций с большими числами, которые не могут быть представлены в памяти компьютера, используется следующий прием. Каждая цифра таких чисел записывается в отдельный элемент массива, и необходимые операции проводятся с элементами массива цифр. Составить программу, выполняющую сложение 20-значных чисел.  **2.3 Результат выполнения задания 3**  **2.3.1** На рисунке 2.5 приведена блок-схема алгоритма для выполнения задания №3.    Рисунок 2.5 - Блок-схема алгоритма задания №3  **2.3.2** Листинг компьютерной программы по заданию 3.  #include <stdio.h>  int mdl(int a)  {  if (a < 0)  a = -a;  return a;  }  int main()  {  int array1[20], array2[20], array3[20], i, flag = 0, option;  printf("Choose an option:\n1) Keyboard input\n2) Initializing an array with random elements\n");  scanf\_s("%d", &option);  switch (option)  {  case 1:  printf("Input the elements of the first array:\n");  for (i = 0; i < 20; i++)  {  scanf\_s("%d", &array1[i]);  }  for (i = 1; i < 20; i++)  {  if (array1[i] < 0)  {  printf("Invalid input");  return 0;  }  }  printf("Input the elements of the second array:\n");  for (i = 0; i < 20; i++)  {  scanf\_s("%d", &array2[i]);  }  for (i = 1; i < 20; i++)  {  if (array2[i] < 0)  {  printf("Invalid input");  return 0;  }  }  break;  case 2:  srand(time(NULL));  for (i = 0; i < 20; i++)  {  array1[i] = rand() % 10;  printf("%d ", array1[i]);  }  printf("\n");  for (i = 0; i < 20; i++)  {  array2[i] = rand() % 10;  printf("%d ", array2[i]);  }  printf("\n");  break;  default:  printf("Error");  return 0;  }  if (array1[0] < 0)  {  for (i = 1; i < 20; i++)  {  array1[i] = -array1[i];  }  flag = 1;  }  if (array2[0] < 0)  {  for (i = 1; i < 20; i++)  {  array2[i] = -array2[i];  }  flag = 1;  }  int temp = 0;  for (i = 19; i >= 0; i--)  {  if (array1[i] + array2[i] + temp >= 10)  {  array3[i] = (array1[i] + array2[i] + temp) % 10;  temp = 1;  }  else  {  array3[i] = array1[i] + array2[i] + temp;  temp = 0;  }  }  if (flag == 1)  {  for (i = 1; i < 20; i++)  {  array3[i] = mdl(array3[i]);  }  }  printf("Answer:\n");  if ((array1[0] + array2[0] + temp) >= 10)  printf("1");  for (i = 0; i <= 19; i++)  printf("%d", array3[i]);  }  **2.3.3** Результат выполнения компьютерной программы в виде «скриншота» изображения на мониторе представлен на рисунке 2.6:   |  |  |  | | --- | --- | --- | |  |  |  | | Рисунок 2.6 - Скриншот результата выполнения компьютерной программы | | | | | | |



## 3 ВЫВОДЫ ПО РЕЗУЛЬТАМ ВЫПОЛНЕНИЯ ЛР

В результате выполнения ЛР изучены методы работы с одномерными массивами, получены практические навыки по написанию алгоритмов на языке C.

# 4 КОНТРОЛЬНЫЕ ВОПРОСЫ И ПРЕДЛОЖЕНИЯ

**4.1** Пояснить основные положения, термины и определения в материалах лекции (лекций) и литературе по теме ЛР.

**4.2** Объяснить алгоритмы выполнения заданий, указанных в данном варианте ЛР.

**4.3** Прокомментировать листинги (фрагменты листингов) компьютерных программ в данном варианте ЛР.

**4.4** Прокомментировать результаты выполнения заданий, указанных в варианте ЛР.