|  |  |
| --- | --- |
| C:\Documents and Settings\admin\Рабочий стол\МИРЭА_ЭМБЛЕМА_приказ.JPG | |
| МИНОБРАНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  «**МОСКОВСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»**  **МИРЭА** | |
| Институт информационных технологий  Кафедра математического обеспечения и стандартизации информационных технологий | |
| **Отчет по лабораторной работе №1** **Тема работы «****Одномерный массив»** | |
| Выполнил: |  |
| Студент группы ИКБО-07-17  Отчет принял:  Преподаватель кафедры МОСИТ | Данилов Д.Е.  Миронов Антон Николаевич |
| Москва 2017 | |

# **Задание 1**

# **Условие задания**

В одномерном массиве А, состоящем из N целых чисел (0<N<1000; 0<A[i]<100) расположить серии повторяющихся элементов в порядке убывания длин серий. Например: 33333336666662222277778845. Подсказка: использовать массив из 99 счетчиков, для каждого возможного значения массива А.

**Декомпозиция**

* Заполнить массив счетчиков
* Выводить набольшее количество элементов(по индексам) из этого массива

# **Реализация функций**

void vvod(int\* arr, int n) //Заполнение массива

{

cout << "Введите элементы массива (от 1 до 99)" << endl;

int a;

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

{

cin >> a;

if (a > 0 && a < 100)

{

arr[i] = a;

continue;

}

else

{

cout << "Введен неверный элемент массива" << endl << "Попробуйте снова" << endl;

system("pause");

break;

}

}

}

void vivod(int\* arr, int n) //Вывод массива

{

cout << "Изначальный массив" << endl;

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

cout << arr[i] << " ";

cout << endl;

}

void autov(int\* arr, int n) //Рандом

{

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

arr[i] = rand() % 99 + 1;

cout << "Полученный массив" << endl;

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

cout << arr[i] << " ";

cout << endl;

system("pause");

}

void sort(int\* arr1, int\* arr2, int n) //Сортировка

{

for (int i = 0; i < 99; i++)

{

arr2[i] = 0;

}

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

{

for (int j = 1; j < 99; j++)

{

if (arr1[i] == j)

arr2[j]++;

}

}

int temp;

int max = 0;

for (int j = 1; j < 99 ; j++)

{

for (int i = 1; i < 99; i++)

{

if (max < arr2[i])

{

max = arr2[i];

temp = i;

}

}

for (int c = 0; c < max; c++)

cout << temp << " ";

max = 0;

arr2[temp] = 0;

}

}

# **Кодирование алгоритма программы**

#include "stdafx.h"

#include <iostream>

#define L 999

#define L1 99

using namespace std;

void vvod(int\* arr, int n) //Заполнение массива

{

cout << "Введите элементы массива (от 1 до 99)" << endl;

int a;

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

{

cin >> a;

if (a > 0 && a < 100)

{

arr[i] = a;

continue;

}

else

{

cout << "Введен неверный элемент массива" << endl << "Попробуйте снова" << endl;

system("pause");

break;

}

}

}

void vivod(int\* arr, int n) //Вывод массива

{

cout << "Изначальный массив" << endl;

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

cout << arr[i] << " ";

cout << endl;

}

void autov(int\* arr, int n) //Рандом

{

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

arr[i] = rand() % 99 + 1;

cout << "Полученный массив" << endl;

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

cout << arr[i] << " ";

cout << endl;

system("pause");

}

void sort(int\* arr1, int\* arr2, int n) //Сортировка

{

for (int i = 0; i < 99; i++)

{

arr2[i] = 0;

}

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

{

for (int j = 1; j < 99; j++)

{

if (arr1[i] == j)

arr2[j]++;

}

}

int temp;

int max = 0;

for (int j = 1; j < 99 ; j++)

{

for (int i = 1; i < 99; i++)

{

if (max < arr2[i])

{

max = arr2[i];

temp = i;

}

}

for (int c = 0; c < max; c++)

cout << temp << " ";

max = 0;

arr2[temp] = 0;

}

}

int main()

{

setlocale(LC\_CTYPE, "RUS");

int N;

int answer1 = 100;

while (answer1 != 0)

{

system("cls");

cout << "Лабораторная работа №1 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

cout << "Меню" << endl;

cout << "1) Задание 1" << endl;

cout << "2) Задание 2" << endl;

cout << "0) Выход" << endl;

cout << "Ваш выбор: ";

cin >> answer1;

system("cls");

cout << "Лабораторная работа №1 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

switch (answer1)

{

case 1: //1 Задание

{

int A2[L1];

int A[L];

int sch = 100;

cout << "Введите длину массива (от 1 до 999)" << endl;

cin >> N;

if (N > 0 && N < 1000)

{

cout << "1) Самостоятельное заполнение" << endl;

cout << "2) Автоматическое заполнение" << endl;

cin >> sch;

switch (sch)

{

case 1:

vvod(A, N);

break;

case 2:

autov(A, N);

break;

default:

break;

}

if (N > 1)

{

if (check(A, N))

{

cout << "Отсортированный массив" << endl;

sort(A, A2, N);

cout << endl;

}

}

memset(A, 0, sizeof(A));

system("pause");

break;

}

else

{

cout << "Введена неправильная длина массива" << endl << "Попробуйте снова" << endl;

system("pause");

break;

}

}

# **Задание 2**

# **Условие задания**

Дан одномерный целочисленный массив А, состоящий из N элементов. N заданное натуральное число.

* Найти количество элементов массива, расположенных между первым минимальным и последним максимальным элементами.
* Выполните сортировку части массива А, между первым минимальным и последним максимальным, располагая элементы в порядке возрастания их значений, методом прямой вставки.
* Удалить из массива А все повторные вхождения элементов, оставив только первое вхождение.
* Найти наибольший общий делитель элементов массива, имеющих индексом простое число.

**Декомпозиция**

* Нахождение первого минимального и последнего максимального элемента массива
* Нахождение количества элементов между ними
* Сортировка части массива между ними методом прямой вставки
* Удаление повторных вхождений
* Проверка индексов массива на простое число
* НОД элементов с индексом – простым числом

# **Реализация функций**

void vvod(int\* arr, int n) //Заполнение массива

{

cout << "Введите элементы массива (от 1 до 99)" << endl;

int a;

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

{

cin >> a;

if (a > 0 && a < 100)

{

arr[i] = a;

continue;

}

else

{

cout << "Введен неверный элемент массива" << endl << "Попробуйте снова" << endl;

system("pause");

break;

}

}

}

void vivod(int\* arr, int n) //Вывод массива

{

cout << "Изначальный массив" << endl;

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

cout << arr[i] << " ";

cout << endl;

}

void autov(int\* arr, int n) //Рандом

{

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

arr[i] = rand() % 99 + 1;

cout << "Полученный массив" << endl;

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

cout << arr[i] << " ";

cout << endl;

system("pause");

}

bool simple(int element) //Простое ли число

{

if (element == 0 || element == 1)

{

return false;

}

else if (element == 2)

{

return true;

}

else {

int calc = 0;

for (int i = 2; i <= element / 2; i++)

{

if (!(element % i))

{

calc++;

}

}

return (calc >= 1) ? false : true;

}

}

int nod(int\* arr, int n) //НОД

{

int temp;

int temp2;

if (n > 2)

{

cout << "Элементы массива, имеющих индексом простое число" << endl;

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

{

if (simple(i))

{

cout << arr[i] << " ";

if (i == 2)

{

temp = arr[i];

}

else

{

temp2 = arr[i];

while (temp != 0 && temp2 != 0)

{

if (temp > temp2)

{

temp = temp % temp2;

}

else

temp2 = temp2 % temp;

}

temp = temp + temp2;

}

}

}

}

else

return 0;

cout << endl;

cout << "НОД элементов массива, имеющих индексом простое число равен" << endl;

return temp;

}

bool in\_array(int a, int\* m, int N) //Подпрограмма для deletep (Есть ли число "a" в массиве)

{

int i;

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

if (m[i] == a)

return true;

return false;

}

void sdv(int a, int\* m, int N) //Подпрограмма для deletep (сдвиг массива влево)

{

int i;

for (i = a; i < N - 1; i++)

m[i] = m[i + 1];

}

void deletep(int\* arr, int n) //Удаление повторяющихся элементов

{

int\* m2;

int i, j = 0;

m2 = new int[n];

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

{

if (in\_array(arr[i], m2, j) == true)

{

sdv(i, arr, n);

n--;

i--;

}

else

m2[j++] = arr[i];

}

delete m2;

cout << "Полученный массив с уникальными элементами" << endl;

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

cout << arr[i] << " ";

cout << endl;

}

bool check(int\* arr, int n) //Проверка для правильной работы сортировки

{

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

if (arr[i] <= 0 || arr[i] >= 100)

return false;

return true;

}

int fmin(int\* arr, int n) //Первый минимальный элемент массива

{

int min = 2147483647;

int i;

int temp;

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

if (arr[i] < min)

{

min = arr[i];

temp = i;

}

return temp;

}

int lmax(int\* arr, int n) //Последний максимальный элемент массива

{

int max = 0;

int i;

int temp;

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

if (arr[i] >= max)

{

max = arr[i];

temp = i;

}

return temp;

}

int kolvo(int\* arr, int n) //Количество элементов массива между lmax и fmin

{

int c = 0;

int i = fmin(arr, n);

int j = lmax(arr, n);

if (i < j)

for (i; i < j - 1; i++)

c++;

else

for (j; j < i - 1; j++)

c++;

cout << "Количество элементов между первым минимальным и последним максимальным" << endl;

return c;

}

void sort2(int\* arr, int n) //Сортировка элементов массива между lmax и fmin

{

int i = fmin(arr, n);

int j = lmax(arr, n);

if (i < j)

{

for (int c = i; c < j; c++)

for (int p = j - 1; p > i + 1; p--)

if (arr[p] < arr[p - 1])

swap(arr[p], arr[p - 1]);

}

else

{

for (int c = j; c < i; c++)

for (int p = i - 1; p < j + 1; p--)

if (arr[p] < arr[p - 1])

swap(arr[p], arr[p - 1]);

}

cout << "Массив после сортировки" << endl;

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

cout << arr[i] << " ";

cout << endl;

}

# **Кодирование алгоритма программы (полная)**

#include "stdafx.h"

#include <iostream>

#define L 999

#define L1 99

using namespace std;

void vvod(int\* arr, int n) //Заполнение массива

{

cout << "Введите элементы массива (от 1 до 99)" << endl;

int a;

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

{

cin >> a;

if (a > 0 && a < 100)

{

arr[i] = a;

continue;

}

else

{

cout << "Введен неверный элемент массива" << endl << "Попробуйте снова" << endl;

system("pause");

break;

}

}

}

void vivod(int\* arr, int n) //Вывод массива

{

cout << "Изначальный массив" << endl;

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

cout << arr[i] << " ";

cout << endl;

}

void autov(int\* arr, int n) //Рандом

{

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

arr[i] = rand() % 99 + 1;

cout << "Полученный массив" << endl;

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

cout << arr[i] << " ";

cout << endl;

system("pause");

}

void sort(int\* arr1, int\* arr2, int n) //Сортировка

{

for (int i = 0; i < 99; i++)

{

arr2[i] = 0;

}

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

{

for (int j = 1; j < 99; j++)

{

if (arr1[i] == j)

arr2[j]++;

}

}

int temp;

int max = 0;

for (int j = 1; j < 99 ; j++)

{

for (int i = 1; i < 99; i++)

{

if (max < arr2[i])

{

max = arr2[i];

temp = i;

}

}

for (int c = 0; c < max; c++)

cout << temp << " ";

max = 0;

arr2[temp] = 0;

}

}

bool simple(int element) //Простое ли число

{

if (element == 0 || element == 1)

{

return false;

}

else if (element == 2)

{

return true;

}

else {

int calc = 0;

for (int i = 2; i <= element / 2; i++)

{

if (!(element % i))

{

calc++;

}

}

return (calc >= 1) ? false : true;

}

}

int nod(int\* arr, int n) //НОД

{

int temp;

int temp2;

if (n > 2)

{

cout << "Элементы массива, имеющих индексом простое число" << endl;

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

{

if (simple(i))

{

cout << arr[i] << " ";

if (i == 2)

{

temp = arr[i];

}

else

{

temp2 = arr[i];

while (temp != 0 && temp2 != 0)

{

if (temp > temp2)

{

temp = temp % temp2;

}

else

temp2 = temp2 % temp;

}

temp = temp + temp2;

}

}

}

}

else

return 0;

cout << endl;

cout << "НОД элементов массива, имеющих индексом простое число равен" << endl;

return temp;

}

bool in\_array(int a, int\* m, int N) //Подпрограмма для deletep (Есть ли число "a" в массиве)

{

int i;

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

if (m[i] == a)

return true;

return false;

}

void sdv(int a, int\* m, int N) //Подпрограмма для deletep (сдвиг массива влево)

{

int i;

for (i = a; i < N - 1; i++)

m[i] = m[i + 1];

}

void deletep(int\* arr, int n) //Удаление повторяющихся элементов

{

int\* m2;

int i, j = 0;

m2 = new int[n];

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

{

if (in\_array(arr[i], m2, j) == true)

{

sdv(i, arr, n);

n--;

i--;

}

else

m2[j++] = arr[i];

}

delete m2;

cout << "Полученный массив с уникальными элементами" << endl;

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

cout << arr[i] << " ";

cout << endl;

}

bool check(int\* arr, int n) //Проверка для правильной работы сортировки

{

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

if (arr[i] <= 0 || arr[i] >= 100)

return false;

return true;

}

int fmin(int\* arr, int n) //Первый минимальный элемент массива

{

int min = 2147483647;

int i;

int temp;

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

if (arr[i] < min)

{

min = arr[i];

temp = i;

}

return temp;

}

int lmax(int\* arr, int n) //Последний максимальный элемент массива

{

int max = 0;

int i;

int temp;

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

if (arr[i] >= max)

{

max = arr[i];

temp = i;

}

return temp;

}

int kolvo(int\* arr, int n) //Количество элементов массива между lmax и fmin

{

int c = 0;

int i = fmin(arr, n);

int j = lmax(arr, n);

if (i < j)

for (i; i < j - 1; i++)

c++;

else

for (j; j < i - 1; j++)

c++;

cout << "Количество элементов между первым минимальным и последним максимальным" << endl;

return c;

}

void sort2(int\* arr, int n) //Сортировка элементов массива между lmax и fmin

{

int i = fmin(arr, n);

int j = lmax(arr, n);

if (i < j)

{

for (int c = i; c < j; c++)

for (int p = j - 1; p > i + 1; p--)

if (arr[p] < arr[p - 1])

swap(arr[p], arr[p - 1]);

}

else

{

for (int c = j; c < i; c++)

for (int p = i - 1; p < j + 1; p--)

if (arr[p] < arr[p - 1])

swap(arr[p], arr[p - 1]);

}

cout << "Массив после сортировки" << endl;

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

cout << arr[i] << " ";

cout << endl;

}

int main()

{

setlocale(LC\_CTYPE, "RUS");

int N;

int answer1 = 100;

while (answer1 != 0)

{

system("cls");

cout << "Лабораторная работа №1 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

cout << "Меню" << endl;

cout << "1) Задание 1" << endl;

cout << "2) Задание 2" << endl;

cout << "0) Выход" << endl;

cout << "Ваш выбор: ";

cin >> answer1;

system("cls");

cout << "Лабораторная работа №1 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

switch (answer1)

{

case 1: //1 Задание

{

int A2[L1];

int A[L];

int sch = 100;

cout << "Введите длину массива (от 1 до 999)" << endl;

cin >> N;

if (N > 0 && N < 1000)

{

cout << "1) Самостоятельное заполнение" << endl;

cout << "2) Автоматическое заполнение" << endl;

cin >> sch;

switch (sch)

{

case 1:

vvod(A, N);

break;

case 2:

autov(A, N);

break;

default:

break;

}

if (N > 1)

{

if (check(A, N))

{

cout << "Отсортированный массив" << endl;

sort(A, A2, N);

cout << endl;

}

}

memset(A, 0, sizeof(A));

system("pause");

break;

}

else

{

cout << "Введена неправильная длина массива" << endl << "Попробуйте снова" << endl;

system("pause");

break;

}

}

case 2: //2 Задание

{

int A2[L1];

int A[L];

int sch = 100;

int sch1 = 100;

cout << "Введите длину массива (от 1 до 999)" << endl;

cin >> N;

if (N > 0 && N < 1000)

{

cout << "1) Самостоятельное заполнение" << endl;

cout << "2) Автоматическое заполнение" << endl;

cin >> sch;

switch (sch)

{

case 1:

{

vvod(A, N);

break;

}

case 2:

{

autov(A, N);

break;

}

default:

break;

}

}

else

{

cout << "Введена неправильная длина массива" << endl << "Попробуйте снова" << endl;

system("pause");

break;

}

{

if (!check(A, N))

break;

}

system("cls");

cout << "Лабораторная работа №1 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

cout << "Меню" << endl;

cout << "1) Количество элементов, расположенных между первым минимальным и последним максимальным элементами" << endl;

cout << "2) Сортировка части массива между первым минимальным и последним максимальным, располагая элементы в порядке возрастания их значений, методом прямой вставки" << endl;

cout << "3) Удаление из массива всех повторных вхождений элементов, оставив только первое вхождение" << endl;

cout << "4) НОД элементов массива, имеющих индексом простое число" << endl;

cout << "0) Выход" << endl;

cout << "Ваш выбор: ";

cin >> sch1;

system("cls");

cout << "Лабораторная работа №1 ИКБО-07-17 Данилов Д.Е. Вариант 7" << endl << endl;

switch (sch1)

{

case 1:

{

vivod(A, N);

cout << kolvo(A, N) << endl;

break;

}

case 2:

{

vivod(A, N);

sort2(A, N);

break;

}

case 3:

{

vivod(A, N);

deletep(A, N);

break;

}

case 4:

{

vivod(A, N);

cout << nod(A, N) << endl;

break;

}

default:

break;

}

system("pause");

break;

}

default:

break;

}

}

system("pause");

return 0;

}