|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|

|  |  |  |
| --- | --- | --- |
|  |    |  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования**«МИРЭА – Российский технологический университет»****РТУ МИРЭА** |

Институт Информационных технологий |
|  |
| Кафедра Математического обеспечения и стандартизации информационных технологий |
|  |
|  |

|  |
| --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2** |
| **по дисциплине** |
| **«**Процедурное программирование**»****Тема: «МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ. ПОДПРОГРАММЫ»** |
|  |
| Выполнил студент группы ИКБО-12-18  | Валяев Д.А. |
| Принял преподаватель | Миронов А.Н. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «28» Февраля 2019 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г. |  *(подпись руководителя)* |

Москва 2018

Задание 1. Работа со статическим одномерным массивом.

Найти максимальное из чисел, встречающееся в массиве более 1 раза.

2. Постановка задачи

Дано: массив Х из М целых чисел.

Ограничения на данные: (0<M<97; -1000<X[i]<100).

Ограничения на решаемую задачу: нет

3. Математическая модель задачи

-m[] – массив

-вводится количество элементов

-вводятся элементы массива

-введенные элементы проверяются на соответствие ограничениям

-поиск максимального элемента массива, который встречается более 1 раза

4. Определение входных и выходных данных

Входные данные-количество элементов ,сами элементы, вводится с клавиатуры.

Результат – максимальный элемент массива, который встречается более 1 раза.

5. Таблица имен объектов программы

|  |  |  |  |
| --- | --- | --- | --- |
| *Имя объекта* | *Диапазон допустимых значений* | *Тип данных* | *Семантика* |
| о | -1000..100 | int | Переменная, аргумент функции |
| g, u, e, y ,h ,max, x | -2 147 483 648 .. 2 147 483 647 | int | Переменная, аргумент функции |
| sk | 97 | int | const |

6. Разработка алгоритма программы



7. Разработка тестов



8. Исходный код программы

/\*Валяев Данила Андреевич, вариант 1, Дан массив Х из М целых чисел (0<M<97; -1000<X[i]<100). Найти максимальное из чисел, встречающееся в данном массиве более 1 раза.

\*/

#include "pch.h"

#include <locale.h>

#include <iostream>

#include <string>

#include <cstdlib>

#include <sstream>

#include <ctime>

#define sk 97

using namespace std;

void mass(int m[], int \*u) {

 int o = 0;

 cout << "Введите кол-во илементов: " << endl;

 cin >> \*u;

 if (\*u > 0 && \*u <= sk) {

 for (int x = 0; x < \*u; x++) {

 cout << "Введите: " << endl;

 cin >> o;

 if (o >= -1000 && o <= 100) {

 m[x] = o;

 }

 for (int x = 0; x < \*u; x++)

 cout << m[x] << " ";

 }

 }

 else

 cout << "error" << endl;

}

void smas(int m[], int \*u) {

 int o = 0;

 cout << "Введите кол-во илементов: " << endl;

 cin >> \*u;

 if (\*u > 0 && \*u < sk) {

 srand(time(NULL));

 for (int x = 0; x < \*u; x++)

 m[x] = 100 - rand() % -1000;

 for (int x = 0; x < \*u; x++)

 cout << m[x] << " ";

 }

 else

 cout << "error" << endl;

}

void search(int m[], int \*u) {

 int y = 0;

 int max = 0;

 for (int h = 0; h < \*u; h++) {

 y = 0;

 for (int e = 0; e < \*u; e++) {

 if (e != h) {

 if (m[e] == m[h]) {

 max = m[e];

 }

 y++;

 }

 }

 }

 cout <<"\n Максимальный элемент: "<< max << endl;

}

int main() {

 cout << "Валяев Данила Андреевич, вариант 1, Дан массив Х из М целых чисел (0<M<97; -1000<X[i]<100). Найти максимальное из чисел, встречающееся в данном массиве более 1 раза." << endl;

 int m[sk];

 int g, u = 0;

 cout << "Массив заполняется рандомно? \n 1-yes \n 2-no" << endl;

 cin >> g;

 if (g == 2) {

 mass(m, &u);

 search(m, &u);

 }

 else if (g == 1) {

 smas(m, &u);

 search(m, &u);

 }

 return 0;

}

Задание 2. Работа с динамическим одномерным массивом.

1. Условие задачи

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

• Cформировать массив В из элементов массива А, являющихся простыми числами.

• Выполнить сортировку части массива В между первым минимальным и последним минимальным значениями методом прямого обмена.

• Удалить все элементы массива, сумма цифр которых кратна 7.

• Найти самую длинную подпоследовательность массива А, состоящую из элементов, которые являются четными числами.

2. Постановка задачи

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

Ограничения на данные: нет.

Ограничения на решаемую задачу: нет.

3. Математическая модель задачи

-Cформировать массив В из элементов массива А, являющихся простыми числами.

-Выполнить сортировку части массива В между первым минимальным и последним минимальным значениями методом прямого обмена.

-Удалить все элементы массива, сумма цифр которых кратна 7.

-Найти самую длинную подпоследовательность массива А, состоящую из элементов, которые являются четными числами.

4. Определение входных и выходных данных

Входные данные – число элементов массива, элементы массива, вводится с клавиатуры.

Результат – массив простых чисел, отсортированный массив, массив без чисел, сумма цифр которых равна 7, максимальная последовательность четных элементов.

5. Таблица имен объектов программы

|  |  |  |  |
| --- | --- | --- | --- |
| *Имя объекта* | *Диапазон допустимых значений* | *Тип данных* | *Семантика* |
| smas, mass, y, n, k, m, n, pr, max, n, st, mk, u, x, g, i, gy, j,counter, o | -2 147 483 648 .. 2 147 483 647 | int | Переменная, аргумент функции |

6. Разработка алгоритма программы



7. Разработка тестов



8. Исходный код программы

/\*Валяев Данила Андреевич, вариант 1 ,Дан одномерный целочисленный массив А, состоящий из N элементов. N заданное натуральное число.

• Cформировать массив В из элементов массива А, являющихся простыми числами.

• Выполнить сортировку части массива В между первым минимальным и последним минимальным значениями методом прямого обмена.

• Удалить все элементы массива, сумма цифр которых кратна 7.

• Найти самую длинную подпоследовательность массива А, состоящую из элементов, которые являются четными числами.\*/

#include "pch.h"

#include <locale.h>

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

void mass(int \*m, int \*n) {

 int o;

 for (int x = 0; x < \*n; x++) {

 cout << "vvedite" << endl;

 cin >> o;

 m[x] = o;

 }

 for (int x = 0; x < \*n; x++)

 cout << m[x] << " ";

}

//-——-----------------------------------—

void smas(int \*m, int \*n) {

 int o = 0;

 srand(time(NULL));

 for (int x = 0; x < \*n; x++)

 m[x] = rand() % 1000;

 for (int x = 0; x < \*n; x++)

 cout << m[x] << " ";

}

//-——-----------------------------------—

void eazy(int \*m, int \*n, int \*jk) {

 int counter = 0;

 int gy = 0;

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

 {

 gy = m[i];

 for (int j = 2; j <= \*n / 2; j++)

 {

 if (m[i] % j == 0 && m[i] != j)

 {

 m[i] = 0;

 counter++;

 break;

 }

 }

 jk[i] = m[i];

 if (jk[i] != 0 && jk[i] != 5)

 cout << jk[i] << " ";

 m[i] = gy;

 }

}

//--------------------------------------—

void sort(int \*n, int \*jk) {

 int ty;

 for (int i = 0; i < \*n - 1; i++) {

 for (int j = 0; j < \*n - i - 1; j++) {

 if (jk[j] > jk[j + 1]) {

 ty = jk[j];

 jk[j] = jk[j + 1];

 jk[j + 1] = ty;

 }

 }

 }

 for (int x = 0; x < \*n; x++) {

 if (jk[x] != 0)

 cout << jk[x] << " ";

 }

}

//--------------------------------------—

int t() {

 int g;

 cout << "Выберите способ заполнения массива \n 1-random \n 2- с клавиатуры" << endl;

 cin >> g;

 return g;

}

//--------------------------------------—

void krat(int \*m, int \*n) {

 int sum, mk = 0;

 for (int u = 0; u < \*n; u++) {

 mk = m[u];

 sum = 0;

 while (m[u] != 0) {

 sum += m[u] % 10;

 m[u] /= 10;

 }

 if (sum % 7 == 0) {

 m[u] = 0;

 }

 else {

 m[u] = mk;

 }

 if (m[u] != 0) {

 cout << m[u] << " ";

 }

 m[u] = mk;

 }

}

//--------------------------------------—

void posled(int \*m, int \*n) {

 int st = 0, max = 0, pr = 0;

 for (int i = 0; i < \*n; i++) {

 if (m[i] % 2 == 0) {

 pr = pr + 1;

 if (i == \*n) {

 if (pr > max) {

 max = pr;

 st = i - pr;

 }

 }

 }

 else {

 if (pr > max) {

 max = pr;

 st = i - pr;

 }

 pr = 0;

 }

 }

 for (int y = st; y < st + max; y++) {

 cout << m[y] << " ";

 }

}

//--------------------------------------—

int count() {

 int k = 0;

 cout << "Введите кол-во элементов массива: ";

 cin >> k;

 return k;

}

//--------------------------------------—

int main() {

 setlocale(LC\_ALL, "RUS");

 cout << "Валяев Данила Андреевич, вариант 1, Дан одномерный целочисленный массив А, состоящий из N элементов. N заданное натуральное число. Cформировать массив В из элементов массива А, являющихся простыми числами.Выполнить сортировку части массива В между первым минимальным и последним минимальным значениями методом прямого обмена.Удалить все элементы массива, сумма цифр которых кратна 7.Найти самую длинную подпоследовательность массива А, состоящую из элементов, которые являются четными числами." << endl;

 int y = t(), h = 0;

 int n = count();

 int \*m = new int(n);

 int \*jk = new int(n);

 if (y == 2) {

 mass(m, &n);

 }

 else if (y == 1) {

 smas(m, &n);

 }

 cout << "\n" << "Массив простых чисел: ";

 eazy(m, &n, jk);

 cout << "\n" << "Отсортированный массив: " << endl;

 sort(&n, jk);

 cout <<"\n" << "Массив без чисел, сумма цифр которых кратна 7: "<<endl;

 krat(m, &n);

 cout <<"\n" << "Максимальная последовательность четных элементов: "<< endl;

 posled(m, &n);

 return 0;

}

Задание 3. Работа с двумерными статическими массивами.

1. Условие задачи

В двумерном массиве храниться информация о количестве учащихся в каждом классе каждой параллели с первой по одиннадцатую. Классы параллели нумеруются буквами А, Б, В, Г. Каждая строка массива хранит информацию о классах одной параллели: в первой строке по первой параллели, во второй по второй и т.д. Определить общее число учащихся в заданной параллели.

2. Постановка задачи

Дано: двумерный статический массив.

Ограничения на данные: нет.

Ограничения на решаемую задачу: нет.

3. Математическая модель задачи

-заполнение массива

-выбор параллели для подсчета учащихся

-вывод количества учащихся в выбранной параллели

4. Определение входных и выходных данных

Входные данные – количество учащихся в каждом классе.

Результат -количество учащихся в выбранной параллели.

5. Таблица имен объектов программы

|  |  |  |  |
| --- | --- | --- | --- |
| *Имя объекта* | *Диапазон допустимых значений* | *Тип данных* | *Семантика* |
| h, o, I, m, y, summ, e,r, s, t,  | -2 147 483 648 .. 2 147 483 647 | int | Переменная, аргумент функции |

6. Разработка алгоритма программы



7. Разработка тестов



8. Исходный код программы

/\*Валяев Данила Андреевич, вариант 1, В двумерном массиве храниться информация о количестве учащихся в каждом классе каждой параллели с первой по одиннадцатую. Классы параллели нумеруются буквами А, Б, В, Г. Каждая строка массива хранит информацию о классах одной параллели: в первой строке по первой параллели, во второй по второй и т.д. Определить общее число учащихся в заданной параллели.\*/

#include "pch.h"

#include <locale.h>

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <string>

using namespace std;

void xx(int m[][4], int n, int y) {

 int h;

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

 for (int o = 0; o < y; o++) {

 cout << "Введите кол-во учащихся в классе: " << endl;

 cin>> h;

 m[i][o] = h;

 }

 }

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

 for (int o = 0; o < y; o++) {

 cout << m[i][o] << " ";

 if (o == 3) {

 cout << "\n";

 }

 }

 }

}

void xxx(int m[][4], int n, int y) {

 int h;

 srand(time(NULL));

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

 for (int o = 0; o < y; o++) {

 m[i][o] = rand() % 30;

 }

 }

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

 for (int o = 0; o < y; o++) {

 cout << m[i][o] << " ";

 if (o == 3) {

 cout << "\n";

 }

 }

 }

}

int par() {

 int y = 0;

 cout << "Выберите параллель для подсчета кол-ва учащихся: " << endl;

 cin>>y;

 if (y > 0 && y <= 11) {

 return y;

 }

 else {

 cout << "error" << endl;

 }

}

int sum(int m[][4], int n, int y, int \*s) {

 int summ = 0;

 int r = \*s - 1;

 for (int e = 0; e < 4; e++) {

 summ = summ + m[r][e];

 }

 return summ;

}

int hi() {

 int y = 0;

 cout << "Способ заполнения массива \n 1-Ввод с клавиатуры \n 2-Random " << endl;

 cin>>y;

 return y;

}

int main() {

 setlocale(LC\_CTYPE, "RUS");

 cout << "Валяев Данила Андреевич, вариант 1, В двумерном массиве храниться информация о количестве учащихся в каждом классе каждой параллели с первой по одиннадцатую. Классы параллели нумеруются буквами А, Б, В, Г. Каждая строка массива хранит информацию о классах одной параллели: в первой строке по первой параллели, во второй по второй и т.д. Определить общее число учащихся в заданной параллели." << endl;

 int t = hi();

 int m[11][4];

 if (t == 1) {

 xx(m, 11, 4);

 }

 else if (t == 2) {

 xxx(m, 11, 4);

 }

 int s = par();

 cout << "Кол-во учащихся в выбранной параллели: "<<sum(m, 11, 4, &s) << endl;

 return 0;

}

Задание 4. Работа с динамическим двумерным массивом.

1. Условие задачи

Дан двухмерный массив. Переформировать его таким образом, чтобы строки располагались по возрастанию их поэлементных сумм.

2. Постановка задачи

Дано: двумерный динамический массив.

Ограничения на данные: нет.

Ограничения на решаемую задачу: нет.

3. Математическая модель задачи

-заполнение массива

-подсчет сумм элементов в строке

-сортировка строк по возрастанию их поэлементный сумм

4. Определение входных и выходных данных

Входные данные – элементы массива.

Результат – отсортированный массив.

5. Таблица имен объектов программы

|  |  |  |  |
| --- | --- | --- | --- |
| *Имя объекта* | *Диапазон допустимых значений* | *Тип данных* | *Семантика* |
| u, k, y ,x ,b, one, max, Mnum, num | -2 147 483 648 .. 2 147 483 647 | int | Переменная, аргумент функции |

6. Разработка алгоритма программы



7. Разработка тестов



8. Исходный код программы

/\*Валяев Данила Андреевич, вариант 1, Дан двухмерный массив. Переформировать его таким образом, чтобы строки располагались по возрастанию их поэлементных сумм.\*/

#include "pch.h"

#include <locale.h>

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <string>

using namespace std;

int ab() {

 int a;

 cout << "Введите кол-во строк: " << endl;

 cin >> a;

 return a;

}

int\*\* tu(int a, int b) {

 int l;

 int \*\*ar = new int\*[a];

 for (int y = 0; y < a; y++) {

 ar[y] = new int[b];

 }

 for (int i = 0; i < a; i++) {

 for (int j = 0; j < b; j++) {

 cout << "Введите число: ";

 cin >> l;

 ar[i][j] = l;

 cout << endl;

 }

 }

 return ar;

}

int ba() {

 int b;

 cout << "Введите кол-во столбцов: " << endl;

 cin >> b;

 return b;

}

int\*\* ty(int a, int b) {

 int \*\*ar = new int\*[a];

 for (int y = 0; y < a; y++) {

 ar[y] = new int[b];

 }

 srand(time(NULL));

 for (int i = 0; i < a; i++) {

 for (int j = 0; j < b; j++) {

 ar[i][j] = rand() % 10;

 }

 }

 return ar;

}

void sort(int \*\*jh, int a, int b) {

 cout << "Исходный массив: " << endl;

 for (int i = 0; i < a; i++) {

 for (int j = 0; j < b; j++) {

 cout << jh[i][j] << " ";

 }

 cout << endl;

 }

 int arr\_buf[20];

 int s\_now = 0, s\_prev = 0;

 for (int p = 0; p < a; p++)

 {

 int\* ptr\_arr = &jh[p][0];

 for (int i = p; i < a; i++)

 {

 for (int j = 0; j < b; j++)

 {

 s\_prev += \*ptr\_arr;

 s\_now += jh[i][j];

 }

 if (s\_prev > s\_now)

 {

 for (int k = 0; k < b; k++)

 {

 arr\_buf[k] = jh[i][k];

 jh[i][k] = \*ptr\_arr;

 \*ptr\_arr = arr\_buf[k];

 ptr\_arr++;

 }

 ptr\_arr -= b;

 }

 s\_now = 0;

 s\_prev = 0;

 }

 }

 cout << "Отсортированный массив: " << endl;

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

 {

 for (int j = 0; j < b; j++)

 cout << jh[i][j] << " ";

 cout << endl;

 }

}

int menu() {

 int k = 0;

 cout << "Заполнение массива: \n 1-рандомно \n 2-ввод с клавиатуры" << endl;

 cin >> k;

 if (k == 1 || k == 2)

 return k;

}

int main() {

 setlocale(LC\_CTYPE, "RUS");

 cout << "Валяев Данила Андреевич, вариант 1, Дан двухмерный массив. Переформировать его таким образом, чтобы строки располагались по возрастанию их поэлементных сумм." << endl;

 int uy = menu(), a = ab(), b = ba();

 if (uy == 1) {

 int \*\*jh = ty(a, b);

 sort(jh, a, b);

 }

 if (uy == 2) {

 int \*\*jh = tu(a, b);

 sort(jh, a, b);

 }

 return 0;

}