Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт Информационных Технологий и Анализ Данных

Кафедра вычислительной техники

**Название работы** – “Работа с командной строкой в C++”

Отчет по лабораторной работе “Работа с командной строкой в C++”

Вариант 25

по дисциплине Информатика

Выполнил

Студент, номер группы ИСТБ-19-2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. Д. Солопов

Дата: \_\_\_\_\_\_\_\_\_\_

Принял

Должность \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Осипова

Дата: \_\_\_\_\_\_\_\_\_\_

Иркутск 2019

**Задача.** Переписать элементы матрицы А в матрицу В, вычитая из каждого элемента строки сумму всех четных (нечетных) чисел этой строки. Организовать два интерфейса ввода: через передачу аргументов в консоль и через пользовательский ввод в консоль

**Схема иерархий функций**

|  |
| --- |
| **argc, argv** |
| **main** |
|  |

|  |
| --- |
| **matrix, line, column** |
| **ShowMatrix** |
|  |

|  |
| --- |
|  |
| **GetUIntValue** |
| **value** |

|  |
| --- |
| **size1, size2, user, rvalue = 200** |
| **GetMatrix** |
| **matrix** |

|  |
| --- |
| **matrix, size1, size2, even** |
| **GetNewMatrix** |
| **B** |

|  |
| --- |
|  |
| **GetIntValue** |
| **value** |

|  |
| --- |
| **Matrix, size1, size2, row, even = true** |
| **SumNumbers** |
| **sum** |

|  |
| --- |
|  |
| **InformationAboutInterface** |
|  |

**Назначение функций**

InformationAboutInterface() – вывод справки о программе

ShowMatrix() – вывод элементов матрицы на консоль.

SumNumbers() – вычисление в определённой строке матрицы суммы чётных(нечётных) элементов.

GetIntValue() – пользовательский ввод целого числа с проверкой на корректность ввода.

GetUIntValue() – пользовательский ввод целого положительного числа с проверкой на корректность ввода.

GetMatrix() – создание матрицы с возможностью как пользовательского ввода элементов, так и рандомного заполнения ими.

GetNewMatrix() – создание матрицы, которая использует элементы другой матрицы для вычисления своих элементов(согласно задаче).

main() – главная функция, которая посредством вызова всех других функций характеризует главный функционал программы.

**Проектирование функций**

**Функция InformationAboutInterface:**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
|  |  |  |  |  |  |

**Алгоритм:**

1. Вывод справки о работе с программой

2. Завершение функции

**Функция SumNumbers:**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| matrix | Матрица размера (size1\*size2) | Целое | sum | Сумма чётных(нечётных) элементов матрицы matrix в строке row | Целое |
| size1 | Количество строк матрицы matrix | Целое > 0 |  |  |  |
| size2 | Количество столбцов матрицы matrix | Целое > 0 |  |  |  |
| row | Строка, в которой нужно искать сумму чётных(нечётных) элементов матрицы matrix | Целое > 0 |  |  |  |
| even | Какие элементы искать? Чётные(true) или нечётные(false) | true/false |  |  |  |

**Алгоритм:**

1. i = 0

2. ПОКА(i < size2) ВЫПОЛНИТЬ

2.1. ЕСЛИ (((matrix[row][i] % 2) РАВНО 0) И (even РАВНО false)) ИЛИ (((matrix[row][i] % 2) НЕ РАВНО 0) И (even РАВНО true)) ТО

2.1.1. sum = sum + matrix[row][i];

2.2. i = i + 1

ПОКА ВСЕ

3. Возврат sum

**Функция ShowMatrix:**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| matrix | Матрица размера (size1\*size2) | Целое |  |  |  |
| line | Количество строк матрицы matrix | Целое > 0 |  |  |  |
| column | Количество столбцов матрицы matrix | Целое > 0 |  |  |  |

**Алгоритм:**

1. i = 0

2. j = 0

3. ПОКА i < line ВЫПОЛНИТЬ

3.1. ПОКА j < column ВЫПОЛНИТЬ

3.1.1. Вывод matrix[i][j]

3.1.2. j = j + 1

ПОКА ВСЕ

3.2. j = 0

3.3. i = i + 1

ПОКА ВСЕ

4. Завершение функции

**Функция GetUIntValue:**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
|  |  |  | value | Целое число большее нуля | Целое > 0 |

**Алгоритм:**

1. Ввод значения в переменную value с проверкой корректности входных данных.

2. Возврат value.

**Функция GetIntValue:**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
|  |  |  | value | Целое число | Целое |

**Алгоритм:**

1. Ввод значения в переменную value с проверкой корректности входных данных.

2. Возврат value.

**Функция GetMatrix:**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| s1 | Количество строк матрицы | Целое > 0 | matrix | Матрица размерности s1\*s2 | Целое число |
| s2 | Количество столбцов матрицы | Целое > 0 |  |  |  |
| user | Вид ввода данных: пользователь(true) или рандомное заполнение(false) | Целое > 0 |  |  |  |
| rvalue | Задание диапазона рандомных чисел | true/false |  |  |  |

**Алгоритм:**

1. Создание матрицы matrix размерностью s1\*s2

2. i = 0

3. k = 0

4. ПОКА i < s1 ВЫПОЛНИТЬ

4.1. ПОКА k < s2 ВЫПОЛНИТЬ

4.1.1. ЕСЛИ user РАВНО true ТО

4.1.1.1. Присваиваем matrix[i][k] число возвращённое функцией GetIntValue():

matrix[i][k] = GetIntValue()

ИНАЧЕ

4.1.1.2. Присваиваем matrix[i][k] рандомное число:

matrix[i][k] = ((rand() % rvalue) - (rvalue/4))

4.1.2. k = k + 1

ПОКА ВСЕ

4.2. k = 0

4.3. i = i + 1

ПОКА ВСЕ

5. Возврат matrix

**Функция GetNewMatrix:**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| matrix | Матрица размера (size1\*size2) | Целое | B | Матрица размера (size1\*size2) | Целое |
| size1 | Количество строк матрицы matrix | Целое > 0 |  |  |  |
| size2 | Количество столбцов матрицы matrix | Целое > 0 |  |  |  |
| even | Сумму каких элементов в каждой строке матрицы matrix нужно искать(true – чётные / false – нечётные) | Целое > 0 |  |  |  |

**Алгоритм:**

1. Создание матрицы B размера size1\*size2 заполненную нулями

2. i = 0

3. k = 0

4. sum = 0

5. ПОКА i < size1 ВЫПОЛНИТЬ

5.1. Присвоить переменой sum значение возвращаемое функцией SumNumbers():

sum = SumNumbers(matrix, size1, size2, i, even)

5.2. ПОКА k < size2 ВЫПОЛНИТЬ

5.3. B[i][k] = (matrix[i][k] – sum)

5.4. k = k + 1

ПОКА ВСЕ

5.3. k = 0

5.4. i = i + 1

ПОКА ВСЕ

6. Возврат B

**Функция main:**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя | Назначение | Диапазон | Имя | Назначение | Диапазон |
| argc | Количество параметров переданное программе через консоль | Целое |  |  |  |
| argv | Массив содержащий параметры переданные через консоль, размер которого равен argc | Строка |  |  |  |

**Алгоритм**

1. size1 = size2 = 0

2. user = false, even = true

3. rvalue = 200

4. ЕСЛИ (argc > 1) И (argv[1] == "-h") ТО

4.1. Вызвать функцию InformationAboutInterface():

InformationAboutInterface();

4.2. Завершить программу

ИНАЧЕ ЕСЛИ (argc > 3) И ((argv[1] РАВНО "-i") ИЛИ (argv[1] РАВНО "-r"))

4.3. ЕСЛИ ((argv[1] РАВНО "-i") И (argv[1] НЕ РАВНО "-r")) ТО

4.3.1. user = true

4.4. ЕСЛИ (argc >= 4) ТО

4.4.1. Присвоить переменной k значение, возвращаемое функцией atoi():

k = atoi(argv[2]);

4.4.2. size1 = k;

4.4.3. Присвоить переменной k значение, возвращаемое функцией atoi():

k = atoi(argv[3]);

4.4.4. size2 = k;

4.4.5. ЕСЛИ (size1 РАВНО 0) ИЛИ (size2 РАВНО 0) ТО

4.4.5.1. Вывести сообщение об ошибке.

4.4.5.2. Завершить программу

4.5. ЕСЛИ (user РАВНО false) И (argc >= 5) ТО

4.5.1. Присвоить переменной k значение, возвращаемое функцией atoi():

k = atoi(argv[4]);

4.5.2. ЕСЛИ (argc >= 6) ТО

4.5.2.1. ЕСЛИ (atoi(argv[5]) РАВНО true) ТО

4.5.2.1.1. even = true

ИНАЧЕ

4.5.2.1.2. even = false

4.5.3. ЕСЛИ (r НЕ РАВНО 1) И (r НЕ РАВНО (-1)) И (r НЕ РАВНО 0)) ТО

4.5.3.1. ЕСЛИ r < 0 ТО

4.5.3.1.1. rvalue = k\*(-1)

ИНАЧЕ

4.5.3.1.2. rvalue = k

ИНАЧЕ

4.5.3.2. even = r

4.6. ЕСЛИ (user НЕ РАВНО false) И (argc >= 5) ТО

4.6.1. ЕСЛИ atoi(argv[4]) НЕ РАВНО 0 ТО

4.6.1.1. even = true

ИНАЧЕ

4.6.1.2. even = false

ИНАЧЕ ЕСЛИ (argc РАВНО 3) ИЛИ (argc > 7) ИЛИ (argc > 1) ТО

4.7. Вывести сообщение об ошибке

4.8. Завершить программу

ИНАЧЕ ЕСЛИ (argc <= 1)

4.9. u = 0

4.10. Присвоить u значение, возвращаемое функцией GetIntValue():

u = GetIntValue();

4.11. ПОКА ((u НЕ РАВНО 0) И ((u НЕ РАВНО 1) И (u НЕ РАВНО 0))) ВЫПОЛНИТЬ

4.11.1. Вывести сообщение об ошибке.

4.11.2. Присвоить u значение, возвращаемое функцией GetIntValue()

ПОКА ВСЕ

4.12. ЕСЛИ u НЕ РАВНО 0 ТО

4.12.1. Вызвать функцию InformationAboutInterface()

4.13. Присвоить переменной size1 возвращаемое значение функции GetUIntValue():

size1 = GetUIntValue();

4.14. Присвоить переменной size2 возвращаемое значение функции GetUIntValue():

Size2 = GetUIntValue();

4.15. u = 0

4.16. Присвоить u значение, возвращаемое функцией GetIntValue():

u = GetIntValue();

4.17. ПОКА ((u НЕ РАВНО 0) И ((u НЕ РАВНО 1) И (u НЕ РАВНО 0))) ВЫПОЛНИТЬ

4.17.1. Вывести сообщение об ошибке.

4.17.2. Присвоить u значение, возвращаемое функцией GetIntValue()

ПОКА ВСЕ

4.18. ЕСЛИ u НЕ РАВНО 0 ТО

4.18.1. user = true

4.19. ЕСЛИ user РАВНО false ТО

4.19.1. u = 0

4.19.2. Присвоить u значение, возвращаемое функцией GetIntValue():

u = GetIntValue();

4.19.3. ПОКА ((u НЕ РАВНО 0) И ((u НЕ РАВНО 1) И (u НЕ РАВНО 0))) ВЫПОЛНИТЬ

4.19.3.1. Вывести сообщение об ошибке.

4.19.3.2. Присвоить u значение, возвращаемое функцией GetIntValue()

ПОКА ВСЕ

4.19.4. ЕСЛИ u НЕ РАВНО 0 ТО

4.19.4.1. Присвоить переменной rvalue значение возвращаемое функцией GetIntValue():

rvalue = GetIntValue();

4.19.4.2. ЕСЛИ rvalue МЕНЬШЕ 0 ТО

4.19.4.2.1. rvalue = (rvalue\*(-1))

ИНАЧЕ ЕСЛИ rvalue РАВНО 0 ТО

4.19.4.2.2. rvalue = 200

4.20. Присвоить u значение, возвращаемое функцией GetIntValue():

u = GetIntValue();

4.21. ПОКА ((u НЕ РАВНО 0) И ((u НЕ РАВНО 1) И (u НЕ РАВНО 0))) ВЫПОЛНИТЬ

4.21.1. Вывести сообщение об ошибке.

4.21.2. Присвоить u значение, возвращаемое функцией GetIntValue()

ПОКА ВСЕ

4.22. ЕСЛИ u НЕ РАВНО 0 ТО

4.22.1. even = true

ИНАЧЕ

4.22.2. even = false

5. Присвоить матрице A значение возвращаемое функцией GetMatrix():

A = GetMatrix(size1, size2, user, rvalue)

6. Присвоить матрице B значение возвращаемое функцией GetNewMatrix():

B = GetNewMatrix(size1, size2, even)

7. Вызвать функцию ShowMatrix(A, size1, size2)

8. Вызвать функцию ShowMatrix(B, size1, size2)

9. Завершить программу.

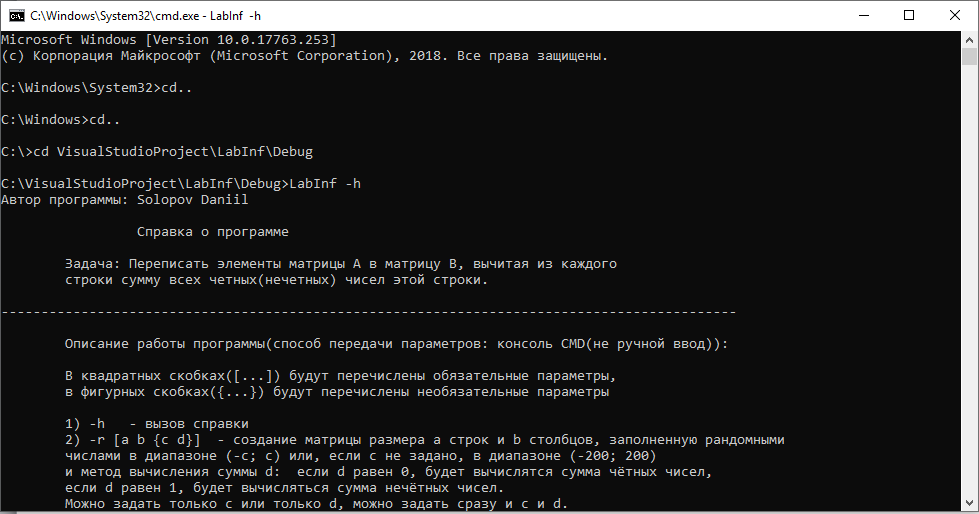
**Приложение к интерфейсу**

Для вызова справки о работе с программой нужно либо передать параметр –h программе через консоль, либо согласиться вызвать справку в начале работы с программой. Программы выполняет задачу. Весь ввод данных контролируется и в случае ошибки пользователю будет выведено сообщение об ошибке. Программа ждёт ввода корректных значений до тех пор, пока пользователь либо не введёт корректное значение, либо не выйдет из программы.

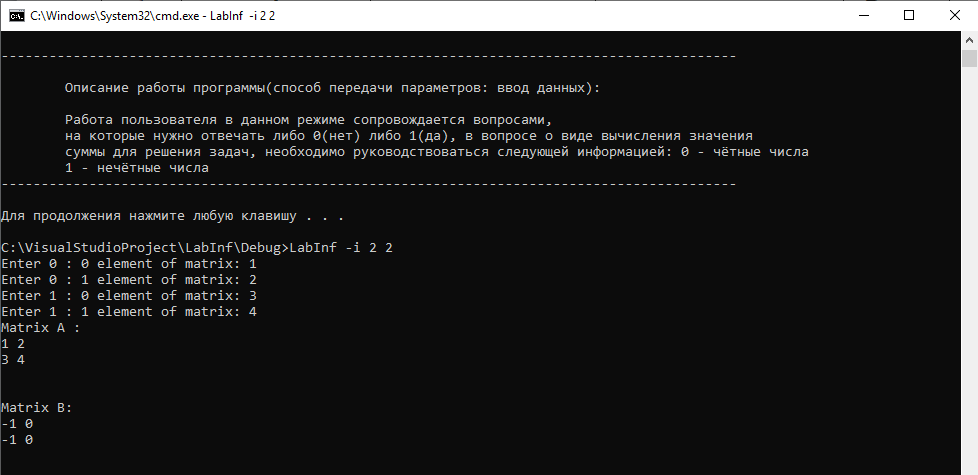
**Тестирование программы**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер теста | Исходные данные | | Результат | Примечания |
|  | Действие | Матрица А | Матрица B |  |
| 1 | -h | - | - | Вызов справки |
| 2 | -i 2 2 | Пользователь сам вводит значения:  1 2  3 4 | -1 0  -1 0 | - |
| 3 | -i 2 av hello wor | - | - | Parameters not correct |
| 4 | -r 2 2 50000 1 | Значения матрицы случайны:  19137 11359  8806 19267 | -11359 -19137  -10461 0 | - |
| 5 | -r 2 3 0 | Значения матрицы случайны:  -30 76 101  177 89 136 | -76 30 55  41 -47 0 | - |
| 6 | -i -i | - | - | Parameters not correct |
| 7 | -r 2 3 100 0 | Значения матрицы случайны:  1 59 29  90 -6 22 | 1 59 29  -16 -112 -84 | - |
| 8 | -i 2 2 abc ed f gf adw f | Пользователь вводит сам:  1 2  3 4 | -1 0  -1 0 | - |
| 9 | -r 1 2 0 1 | Значения матрицы случайны:  4 155 | 0 151 | - |

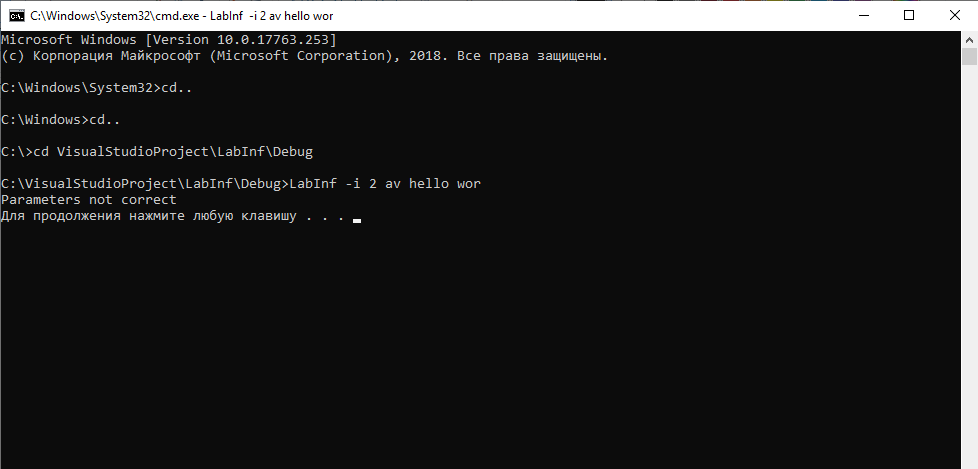
**Тест 1**

****

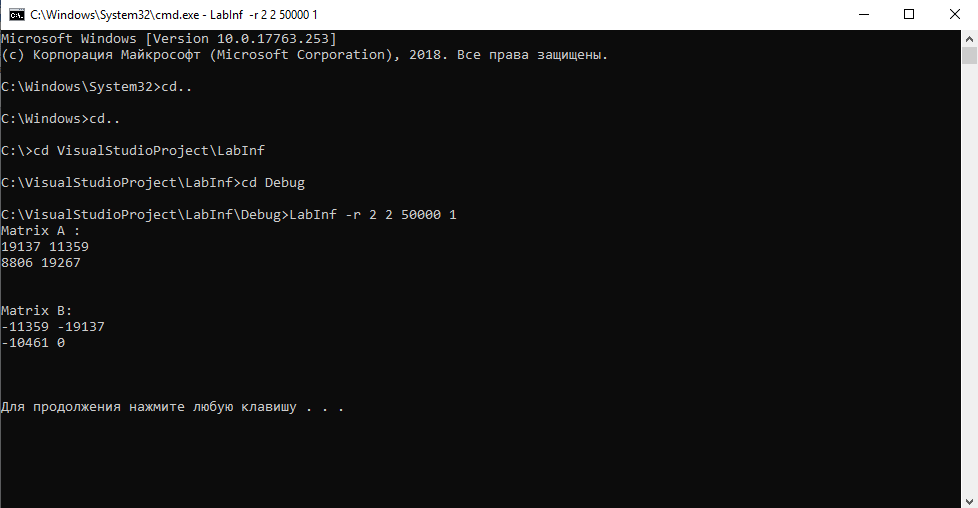
**Тест 2**

****

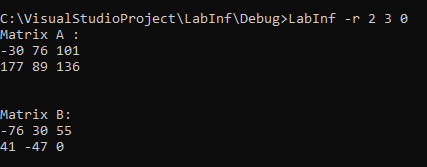
**Тест 3**

****

**Тест 4**

****

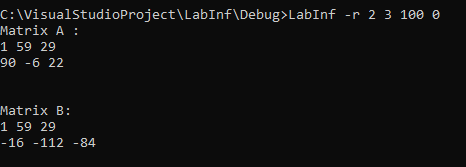
**Тест 5**

****

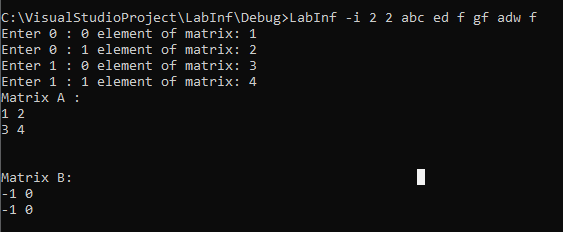
**Тест 6**

****

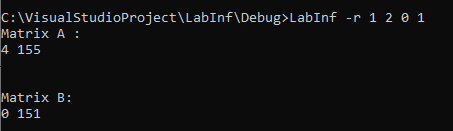
**Тест 7**

****

**Тест 8**

****

**Тест 9**

****

Программный код

#include <iostream>

#include <ctime>

#include <limits>

#include <string>

/\*Переписать элементы матрицы А в матрицу В, вычитая из каждого

элемента строки сумму всех четных (нечетных) чисел этой строки.\*/

int SumNumbers(int\*\* matrix, size\_t size1, size\_t size2, size\_t row, bool even = true) { // сумма всех чётных(нечётных) чисел row строки

if ((matrix == nullptr) || (size1 <= row))

return 0;

int sum = 0;

for (size\_t i = 0; i < size2; i++)

if ((!(matrix[row][i] % 2) && (!even))

|| ((matrix[row][i] % 2) && (even)))

sum += matrix[row][i];

return sum;

}

int\*\* GetMatrix(size\_t, size\_t, bool, unsigned int); //прототип(для вызова в GetNewMatrix)

int\*\* GetNewMatrix(int\*\* matrix, size\_t size1, size\_t size2, bool even = true) { //получение новой матрицы соответственно условию задачи

if ((matrix == nullptr) || (!size1) || (!size2))

return nullptr;

int\*\* B = GetMatrix(size1, size2, false, 1); //Создаём матрицу с заполнением нулями

int sum = 0;

for (size\_t i = 0; i < size1; i++) {

sum = SumNumbers(matrix, size1, size2, i, even); //Запись суммы всех чётных элементов i строки в переменную sum

for (size\_t k = 0; k < size2; k++) {

B[i][k] = (matrix[i][k] - sum);

}

}

return B;

}

size\_t GetUIntValue() { //возвращает число, характеризующее размер

int value = 0;

while (!(std::cin >> value) || (value <= 0) || (std::cin.fail())) {

std::cout << "Please enter correct value: ";

std::cin.clear();

std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

value = 0;

}

std::cin.clear();

std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

return (size\_t)value;

}

int GetIntValue() {

int value;

while (!(std::cin >> value) || (std::cin.fail())) {

std::cout << "Please enter correct value: ";

std::cin.clear();

std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

}

std::cin.clear();

std::cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

return value;

}

int\*\* GetMatrix(size\_t s1, size\_t s2, bool user, unsigned int rvalue = 200) {

if (!user)

srand(time(nullptr));

int\*\* matrix = new int\* [s1];

for (size\_t i = 0; i < s1; i++) {

matrix[i] = new int[s2];

for (size\_t k = 0; k < s2; k++) {

if(user)

std::cout << "Enter " << i << " : " << k << " element of matrix: ";

matrix[i][k] = (user) ? GetIntValue() : ((rand() % (int)(rvalue)) - ((rvalue/4)\*(rand() % 2)));

}

}

return matrix;

}

void ShowMatrix(int\*\* matrix, size\_t line, size\_t column) {

if ((matrix == nullptr) || (!line) || (!column))

return;

for (size\_t i = 0; i < line; i++) {

for (size\_t k = 0; k < column; k++)

std::cout << matrix[i][k] << ' ';

std::cout << std::endl;

}

}

void InformationAboutInterface() {

setlocale(LC\_ALL, "RUS");

std::cout << "Автор программы: Solopov Daniil\n\n";

std::cout << "\t\t Справка о программе\n\n";

std::cout << "\tЗадача: Переписать элементы матрицы А в матрицу В, вычитая из каждого\n";

std::cout << "\tстроки сумму всех четных(нечетных) чисел этой строки.\n\n";

std::cout << "--------------------------------------------------------------------------------------------\n\n";

std::cout << "\tОписание работы программы(способ передачи параметров: консоль CMD(не ручной ввод)):\n\n";

std::cout << "\tВ квадратных скобках([...]) будут перечислены обязательные параметры,\n";

std::cout << "\tв фигурных скобках({...}) будут перечислены необязательные параметры\n\n";

std::cout << "\t1) -h - вызов справки\n";

std::cout << "\t2) -r [a b {c d}] - создание матрицы размера a строк и b столбцов, заполненную рандомными\n";

std::cout << "\tчислами в диапазоне (-c; c) или, если c не задано, в диапазоне (-200; 200)\n";

std::cout << "\tи метод вычисления суммы d: если d равен 0, будет вычислятся сумма чётных чисел,\n";

std::cout << "\tесли d равен 1, будет вычисляться сумма нечётных чисел.\n";

std::cout << "\tМожно задать только c или только d, можно задать сразу и c и d.\n";

std::cout << "\tПричем, если c будет в диапазоне [-1;1], то это будет как d, и d в таком случае, если он задан\n";

std::cout << "\tбудет проигнорирован. Если d не задан или задан ошибочно(символы), то вычисляться будет сумма чётных чисел.\n";

std::cout << "\t3) -i [a b {d}] - создание матрицы размера a строк и b столбцов, заполняемую пользователем\n";

std::cout << "\tсамостоятельно. Параметр d характеризует способ вычисление суммы: 0 - сумма чётных чисел,\n";

std::cout << "\t1 - сумма нечётных чисел. Если параметр d введён ошибочно(символы), то вычисляться будет\n";

std::cout << "\tсумма чётных чисел. \n\n";

std::cout << "--------------------------------------------------------------------------------------------\n\n";

std::cout << "\tОписание работы программы(способ передачи параметров: ввод данных):\n\n";

std::cout << "\tРабота пользователя в данном режиме сопровождается вопросами,\n";

std::cout << "\tна которые нужно отвечать либо 0(нет) либо 1(да), в вопросе о виде вычисления значения\n";

std::cout << "\tсуммы для решения задач, необходимо руководствоваться следующей информацией: 0 - чётные числа\n";

std::cout << "\t1 - нечётные числа\n";

std::cout << "--------------------------------------------------------------------------------------------\n\n";

}

int main(int argc, char\* argv[]){

size\_t size1 = 0, size2 = 0;

int rvalue = 200;

bool user = false, even = false;

if ((argc > 1) && (std::string(argv[1]).data() == std::string("-h"))) {

InformationAboutInterface();

system("PAUSE");

return 1;

}

else if ((argc > 3) && ((std::string(argv[1]).data() == std::string("-i"))

|| (std::string(argv[1]).data() == std::string("-r")))) {

user = (std::string(argv[1]).data() == std::string("-i"))

&& (!(std::string(argv[1]).data() == std::string("-r")));

if (argc >= 4) {

int k = atoi(argv[2]);

size1 = (k < 0) ? (k \* (-1)) : k;

k = atoi(argv[3]);

size2 = (k < 0) ? (k \* (-1)) : k;

if ((!size1) || (!size2)) {

std::cout << "Parameters not correct\n";

system("PAUSE");

return 1;

}

}

if ((!user) && (argc >= 5)) {

int r = atoi(argv[4]);

if (argc >= 6)

even = atoi(argv[5]);

if ((r != 1) && (r != (-1)) && (r))

rvalue = (r < 0) ? (-1) \* r : r;

else

even = r;

}

if((user) && (argc >= 5))

even = atoi(argv[4]);

}

else if ((argc == 3) || (argc > 7)) {

std::cout << "Parameters not correct\n";

system("PAUSE");

return 1;

}

else if(argc <= 1){

int u = 0;

std::cout << "Call information about the interface?(1/0): ";

while ((u = GetIntValue()) && ((u != 1) && (u != 0)))

std::cout << "Please enter correct value: ";

if (u)

InformationAboutInterface();

std::cout << "Enter size of matrix(line): ";

size1 = GetUIntValue();

std::cout << "Enter size of matrix(column): ";

size2 = GetUIntValue();

u = 0;

std::cout << "Enter data to user or not?(1/0): ";

while ((u = GetIntValue()) && ((u != 1) && (u != 0)))

std::cout << "Please enter correct value: ";

if (u)

user = true;

if (!user) {

std::cout << "Enter random diapazon to user or not?(1/0): ";

u = 0;

while ((u = GetIntValue()) && ((u != 1) && (u != 0)))

std::cout << "Please enter correct value: ";

if (u) {

std::cout << "Enter random diapazon: ";

rvalue = GetIntValue();

if (rvalue <= 0)

rvalue = (!rvalue) ? 200 : ((-1) \* rvalue);

}

}

std::cout << "Enter even/not even numbers(1/0): ";

u = 0;

while ((u = GetIntValue()) && ((u != 1) && (u != 0)))

std::cout << "Please enter correct value: ";

if (u)

even = false;

else

even = true;

system("cls");

}

int\*\* A = nullptr, \*\* B = nullptr;

A = GetMatrix(size1, size2, user, rvalue);

B = GetNewMatrix(A, size1, size2, even);

std::cout << "Matrix A : \n";

ShowMatrix(A, size1, size2);

std::cout << "\n\nMatrix B: \n";

ShowMatrix(B, size1, size2);

std::cout << std::endl;

for (size\_t i = 0; i < size1; i++) {

delete[] A[i];

delete[] B[i];

}

delete[] A;

delete[] B;

std::cout << "\n\n";

system("PAUSE");

return 0;

}