Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**Отчёт**

по лабораторной работе №1

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Простые структуры данных»

Выполнил:

 студент группы 20ВВ2

Лазутин Д.Д.

 Принял:

 к.т.н., доцент Юрова О.В.

Пенза 2020

**Задание 1**: написать программу, вычисляющую разницу между максимальным и минимальным элементами массива.

**Задание 2**: написать программу, реализующую инициализацию массива случайными числами.

**Задание 3**: написать программу, реализующую создание массива произвольного размера, вводимого с клавиатуры.

**Задание 4**: написать программу, вычисляющую сумму значений в каждом столбце (или строке) двумерного массива.

*Примечание: задания 1-4 выполнены в одной программе.*

**Описание метода решения:**

Объявлены указатели A (массив) и M (матрица), переменные S (размер массива), M\_i и M\_j (размеры матрицы), счётчики i и j, Min и Max (мин. и макс. элементы массива А), min\_i и max\_i (индексы мин. и макс. элемента массива соответственно).

Вводим размеры массива и матрицы. Для них выделяется память, затем происходит заполнение массива и матрицы (построчно) случайными числами.

Для массива А ищем минимальный и максимальный элемент, затем выводим на экран их разность.

Для матрицы М выбираем, каким способом суммировать её элементы – по строкам или по столбцам. Затем производится каждая сумма элементов по выбранному способу и выводится на экран.

Освобождаем выделенную под массив и матрицу память.

**Листинг** LZ\_1\_1-4.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <locale.h>

void main()

{

int \*A=NULL, S=0, Min, Max, min\_i=0, max\_i=0; //Массив, его размер, значения макс. и мин. элементов и соответствующие им индексы.

int \*\*M=NULL, M\_i, M\_j, i=0, j=0, Sum; //Матрица, кол-во столбцов и строк, счётчики и переменная под сумму элементов.

int Choice; //Выбор способа суммирования элементов матрицы (по столбцам / по строкам)

srand(time(NULL)); //Инициализатор датчика случайных чисел

setlocale(LC\_ALL, "Rus");

printf("Введите размер массива: ");

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

while (S <= 0)

{

printf("Размер массива задан некорректно! Введите целое положительное число: ");

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

}

printf("Введите количество столбцов матрицы: ");

scanf\_s("%d", &M\_i);

while (M\_i <= 0)

{

printf("Количество столбцов матрицы задано некорректно! Введите целое положительное число: ");

scanf\_s("%d", &M\_i);

}

printf("Введите количество строк матрицы: ");

scanf\_s("%d", &M\_j);

while (M\_j <= 0)

{

printf("Количество строк матрицы задано некорректно! Введите целое положительное число: ");

scanf\_s("%d", &M\_j);

}

A = (int\*)malloc(S \* sizeof(int)); //Выделение памяти под массив

M=(int\*\*)malloc(M\_j\*sizeof(int)); //Выделение памяти под указатели на строки матрицы

printf("\nМассив A: ");

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

{

\*(A + i)=rand()%200-100; //Заполнение массива случайными числами

printf("%5d", \*(A+i));

}

Min = \*A; Max = \*A;

for (i = 0; i < S; i++) //Поиск мин. и макс. элементов массива

{

if (\*(A + i) < Min)

{

Min = \*(A + i);

min\_i = i;

}

if (\*(A + i) > Max)

{

Max = \*(A + i);

max\_i = i;

}

}

printf("\nМакс. элемент - Мин. элемент = M[%d] - M[%d] = %d - %d = %d\n", max\_i, min\_i, Max, Min, Max-Min);

printf("Матрица M:\n");

for (j = 0; j < M\_j; j++)

{

\*(M+j)=(int\*)malloc(M\_i\*sizeof(int));

for (i = 0; i < M\_i; i++)

{

\*(\*(M+j)+i) = rand() % 200 - 100;

printf("%5d", \*(\*(M+j)+i));

}

printf("\n");

}

printf("Выберите способ суммирования элементов матрицы:\n1-По строкам\n2-По столбцам\n");

scanf("%d", &Choice);

while((Choice!=1)&&(Choice!=2))

{

printf("Некорректно введённое число! Введите 1 или 2.\n");

scanf("%d", &Choice);

}

switch(Choice)

{

case(1): //Сумма элементов по строкам

{

for(j=0;j<M\_j;j++)

{

for(i=0, Sum=0;i<M\_i;i++)

Sum+=\*(\*(M+j)+i);

printf("Сумма по %d-ой строке=%d\n", j+1, Sum);

}

break;

}

case(2): //Сумма элементов по столбцам

{

for(i=0;i<M\_i;i++)

{

for(j=0, Sum=0;j<M\_j;j++)

Sum+=\*(\*(M+j)+i);

printf("Сумма по %d-му столбцу=%d\n", i+1, Sum);

}

break;

}

default:

break;

}

free(M);

free(A);

}

**Задание 5**: написать программу, осуществляющую поиск среди структур student структуру с заданными параметрами (фамилией, именем и т.д.).

**Описание метода решения:**

Введён шаблон структуры Student. Объявлены динамическая структура stud по тому же шаблону, переменные Size (кол-во структур), i (счётчик), Choice (выбор – искать нужные структуры или нет), Area (выбор поля структуры, по которому будет вестись поиск).

Вводим размер «базы данных» (кол-во записей о студентах), выделяем stud память, вводим информацию о каждом студенте. Затем все записи выводятся на экран.

Далее мы выбираем, начинать поиск записей или нет. Если выберем «Нет», то программа завершает работу. Если выбрать «Да», программа предложит, по какому критерию искать записи (имя, фамилия, факультет, номер зачётки). Выбираем один из вариантов, а потом вводим слово (или число), по которому будет вестись поиск. Далее вызывается функция Search, которая ищет записи по введённым ранее данным и выводит их на экран. Если же ничего не будет найдено, то появится сообщение об этом.

В конце освобождается память, выделенная stud, и программа завершает работу.

**Листинг** LZ\_1\_5.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <locale.h>

#include <stdlib.h>

#include <string.h>

typedef struct Student

{

char Name[20];

char Fam[20];

char Facult[20];

int Zach\_Num;

}\*Base;

void Search(int Area, Base Str, int S)

{

int i, Z, k=0;

char Buf[20];

switch(Area)

{

case(1):

{

printf("Искать по фамилии: "); scanf("%s", &Buf);

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

if (!strcmp(Str[i].Fam, Buf))

{

printf("%d| Студент %s %s обучается на факультете %s, номер зачётной книжки: %d\n", i+1, Str[i].Fam, Str[i].Name, Str[i].Facult, Str[i].Zach\_Num);

k++;

}

break;

}

case(2):

{

printf("Искать по имени: "); scanf("%s", &Buf);

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

if (!strcmp(Str[i].Name, Buf))

{

printf("%d| Студент %s %s обучается на факультете %s, номер зачётной книжки: %d\n", i+1, Str[i].Fam, Str[i].Name, Str[i].Facult, Str[i].Zach\_Num);

k++;

}

break;

}

case(3):

{

printf("Искать по факультету: "); scanf("%s", &Buf);

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

if (!strcmp(Str[i].Facult, Buf))

{

printf("%d| Студент %s %s обучается на факультете %s, номер зачётной книжки: %d\n", i+1, Str[i].Fam, Str[i].Name, Str[i].Facult, Str[i].Zach\_Num);

k++;

}

break;

}

case(4):

{

printf("Искать по номеру зачётной книжки: "); scanf("%d", &Z);

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

if (Str[i].Zach\_Num==Z)

{

printf("%d| Студент %s %s обучается на факультете %s, номер зачётной книжки: %d\n", i+1, Str[i].Fam, Str[i].Name, Str[i].Facult, Str[i].Zach\_Num);

k++;

}

break;

}

}

if(k==0)

printf("Ничего не найдено по введённому запросу...\n\n");

}

int main()

{

int i=0, Size=0, Choice, Area;

Base stud=NULL;

setlocale(LC\_ALL, "Rus");

printf("Сколько студентов должно быть записано в базу данных?\n");

scanf("%d", &Size);

while(Size<=0)

{

printf("Число введено некорректно! Введите целое положительное число: ");

scanf("%d", &Size);

}

stud=malloc(Size\*sizeof(struct Student));

if(stud==NULL)

{

printf("Ошибка выделения памяти!\n");

return -1;

}

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

{

printf("\nВведите данные о %d-ом студенте...\n", i+1);

printf("Фамилия: "); scanf("%s", &(stud[i].Fam));

printf("Имя: "); scanf("%s", &(stud[i].Name));

printf("На каком факультете студент обучается: "); scanf("%s", &(stud[i].Facult));

printf("Номер зачётной книжки: "); scanf("%d", &(stud[i].Zach\_Num));

}

printf("\n\n");

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

{

printf("%d| Студент %s %s обучается на факультете %s, номер зачётной книжки: %d\n", i+1, stud[i].Fam, stud[i].Name, stud[i].Facult, stud[i].Zach\_Num);

}

printf("\nЖелаете ли Вы найти информацию о конкретном студенте? (0-Нет, 1-Да)\n");

scanf("%d", &Choice);

while((Choice!=0) && (Choice!=1))

{

printf("Ответ введён некорректно! Введите цифру 0 или 1:\n");

scanf("%d", &Choice);

}

switch(Choice)

{

case(0):

{

printf("Завершение работы программы...\n");

break;

}

case(1):

{

printf("По какому критерию Вы хотите найти студента?");

printf("\n1-Фамилия, 2-Имя, 3-Факультет, 4-Номер зачётной книжки\n");

scanf("%d", &Area);

while((Area!=1)&&(Area!=2)&&(Area!=3)&&(Area!=4))

{

printf("Ответ введён некорректно! Введите цифру от 1 до 4:\n");

scanf("%d", &Area);

}

Search(Area, stud, Size);

break;

}

}

free(stud);

return 0;

}

**Результаты работы программ**

Результаты работы программы для 1-4-го заданий представлены на рис.1 и рис.2.

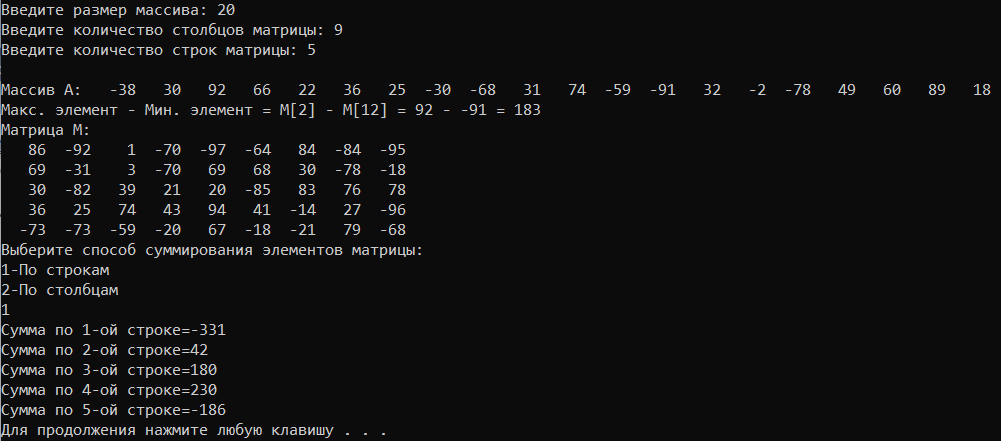


Рис.1.

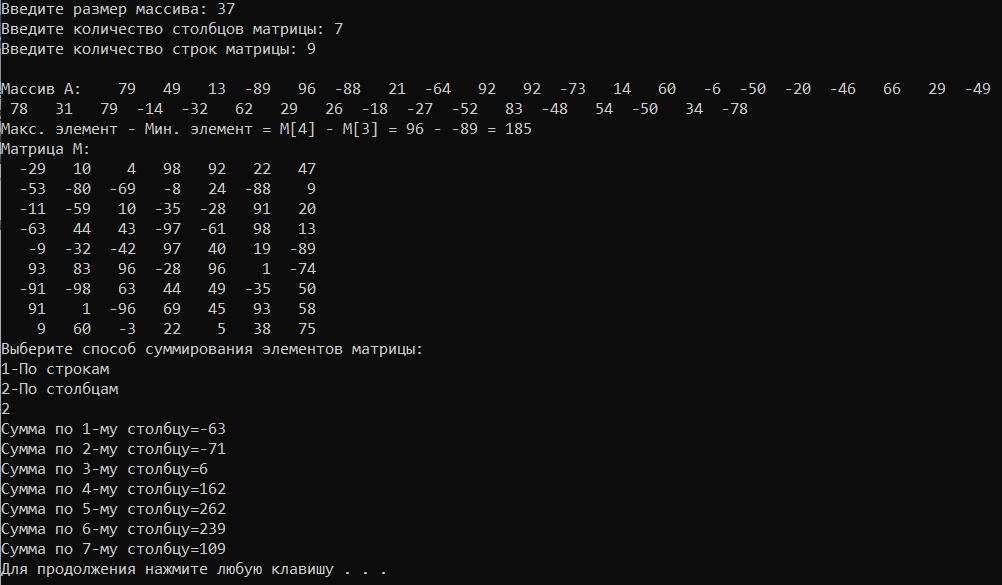


Рис.2.

Результат работы программы для 5-го задания представлен на рис.3.

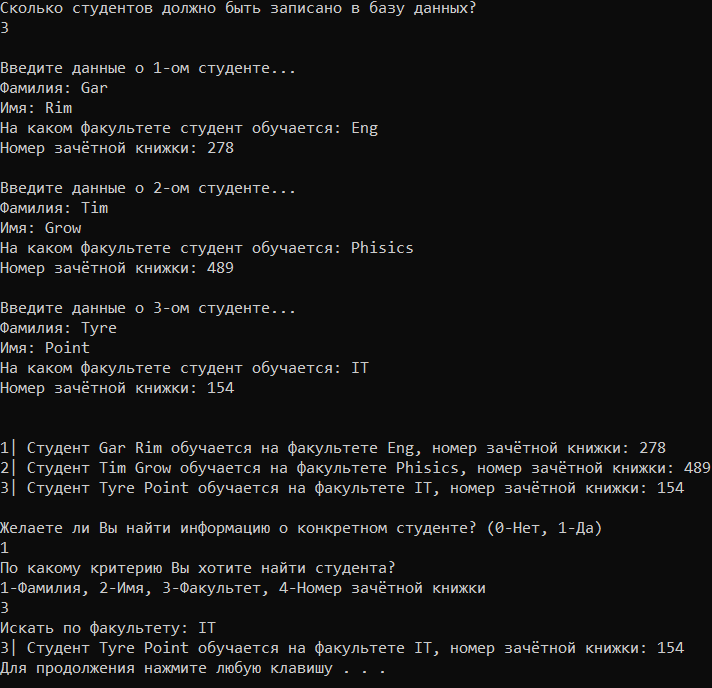


Рис.3.