Министерство науки и высшего 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»

Вариант 1

Выполнили:

Студенты группы 24ВВВ1

Захаров А.В.

Гурин А.Н.

Приняли:

к.т.н., доцент Юрова О.В.

Пенза 2025**Лабораторное задание**

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.

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

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS #include <stdio.h> #include <stdlib.h> #include <time.h> #include <locale.h> #include <string.h>

struct student { char famil[20]; char name[20]; char facult[20]; int Nomzach; };

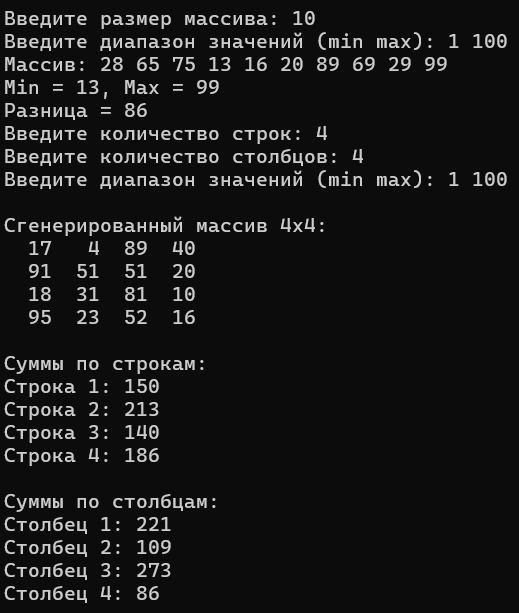
int main() { system("chcp 1251 > nul"); setlocale(LC\_ALL, "Ru"); srand(time(NULL));

int size;  
printf("Введите размер массива: ");  
scanf("%d", &size);  
  
if (size <= 0) {  
 printf("Ошибка: размер массива должен быть положительным!\n");  
 return 1;  
}  
  
int minA, maxA;  
printf("Введите диапазон значений (min max): ");  
scanf("%d %d", &minA, &maxA);  
  
if (minA > maxA) {  
 printf("Ошибка: минимальное значение не может быть больше максимального!\n");  
 return 1;  
}  
  
int\* array = (int\*)malloc(size \* sizeof(int));  
  
for (int i = 0; i < size; i++) {  
 array[i] = rand() % (maxA - minA + 1) + minA;  
}  
  
int min = array[0], max = array[0];  
for (int i = 1; i < size; i++) {  
 if (array[i] < min) min = array[i];  
 if (array[i] > max) max = array[i];  
}  
  
printf("Массив: ");  
for (int i = 0; i < size; i++) {  
 printf("%d ", array[i]);  
}  
printf("\n");  
  
printf("Min = %d, Max = %d\n", min, max);  
printf("Разница = %d\n", max - min);  
  
free(array);  
  
int rows, cols;  
int minM, maxM;  
  
printf("Введите количество строк: ");  
scanf("%d", &rows);  
printf("Введите количество столбцов: ");  
scanf("%d", &cols);  
  
printf("Введите диапазон значений (min max): ");  
scanf("%d %d", &minM, &maxM);  
  
if (rows <= 0 || cols <= 0 || minM > maxM) {  
 printf("Ошибка ввода данных!\n");  
 return 1;  
}  
  
int\*\* matrix = (int\*\*)malloc(rows \* sizeof(int\*));  
  
for (int i = 0; i < rows; i++) {  
 matrix[i] = (int\*)malloc(cols \* sizeof(int));  
 if (matrix[i] == NULL) {  
 printf("Ошибка выделения памяти для строки %d!\n", i);  
 for (int j = 0; j < i; j++) {  
 free(matrix[j]);  
 }  
 free(matrix);  
 return 1;  
 }  
}  
  
for (int i = 0; i < rows; i++) {  
 for (int j = 0; j < cols; j++) {  
 matrix[i][j] = rand() % (maxM - minM + 1) + minM;  
 }  
}  
  
printf("\nСгенерированный массив %dx%d:\n", rows, cols);  
for (int i = 0; i < rows; i++) {  
 for (int j = 0; j < cols; j++) {  
 printf("%4d", matrix[i][j]);  
 }  
 printf("\n");  
}  
  
printf("\nСуммы по строкам:\n");  
for (int i = 0; i < rows; i++) {  
 int row\_sum = 0;  
 for (int j = 0; j < cols; j++) {  
 row\_sum += matrix[i][j];  
 }  
 printf("Строка %d: %d\n", i + 1, row\_sum);  
}  
  
printf("\nСуммы по столбцам:\n");  
for (int j = 0; j < cols; j++) {  
 int col\_sum = 0;  
 for (int i = 0; i < rows; i++) {  
 col\_sum += matrix[i][j];  
 }  
 printf("Столбец %d: %d\n", j + 1, col\_sum);  
}  
  
for (int i = 0; i < rows; i++) {  
 free(matrix[i]);  
}  
free(matrix);  
  
int numStudents;  
printf("Введите количество студентов: ");  
scanf("%d", &numStudents);  
  
struct student\* students = (struct student\*)malloc(numStudents \* sizeof(struct student));  
  
for (int i = 0; i < numStudents; i++) {  
 printf("\nСтудент %d:\n", i + 1);  
 printf("Введите фамилию: ");  
 scanf("%19s", students[i].famil);  
 printf("Введите имя: ");  
 scanf("%19s", students[i].name);  
 printf("Введите факультет: ");  
 scanf("%19s", students[i].facult);  
 printf("Введите номер зачетной книжки: ");  
 scanf("%d", &students[i].Nomzach);  
}  
  
int searchChoice;  
do {  
 printf("\nВыберите параметр для поиска:\n");  
 printf("1 - По фамилии\n");  
 printf("2 - По имени\n");  
 printf("3 - По факультету\n");  
 printf("4 - По номеру зачетной книжки\n");  
 printf("0 - Выход\n");  
 printf("Ваш выбор: ");  
 scanf("%d", &searchChoice);  
  
 if (searchChoice == 0) break;  
  
 char searchStr[20];  
 int searchNum;  
 int found = 0;  
  
 switch (searchChoice) {  
 case 1:  
 printf("Введите фамилию для поиска: ");  
 scanf("%19s", searchStr);  
 printf("\nРезультаты поиска по фамилии '%s':\n", searchStr);  
 for (int i = 0; i < numStudents; i++) {  
 if (strcmp(students[i].famil, searchStr) == 0) {  
 printf("Студент: %s %s, Факультет: %s, Номер зачетки: %d\n",  
 students[i].famil, students[i].name,  
 students[i].facult, students[i].Nomzach);  
 found = 1;  
 }  
 }  
 break;  
 case 2:  
 printf("Введите имя для поиска: ");  
 scanf("%19s", searchStr);  
 printf("\nРезультаты поиска по имени '%s':\n", searchStr);  
 for (int i = 0; i < numStudents; i++) {  
 if (strcmp(students[i].name, searchStr) == 0) {  
 printf("Студент: %s %s, Факультет: %s, Номер зачетки: %d\n",  
 students[i].famil, students[i].name,  
 students[i].facult, students[i].Nomzach);  
 found = 1;  
 }  
 }  
 break;  
 case 3:  
 printf("Введите факультет для поиска: ");  
 scanf("%19s", searchStr);  
 printf("\nРезультаты поиска по факультету '%s':\n", searchStr);  
 for (int i = 0; i < numStudents; i++) {  
 if (strcmp(students[i].facult, searchStr) == 0) {  
 printf("Студент: %s %s, Факультет: %s, Номер зачетки: %d\n",  
 students[i].famil, students[i].name,  
 students[i].facult, students[i].Nomzach);  
 found = 1;  
 }  
 }  
 break;  
 case 4:  
 printf("Введите номер зачетной книжки для поиска: ");  
 scanf("%d", &searchNum);  
 printf("\nРезультаты поиска по номеру зачетной книжки '%d':\n", searchNum);  
 for (int i = 0; i < numStudents; i++) {  
 if (students[i].Nomzach == searchNum) {  
 printf("Студент: %s %s, Факультет: %s, Номер зачетки: %d\n",  
 students[i].famil, students[i].name,  
 students[i].facult, students[i].Nomzach);  
 found = 1;  
 }  
 }  
 break;  
 default:  
 printf("Неверный выбор!\n");  
 continue;  
 }  
  
 if (!found) {  
 printf("Студенты с указанными параметрами не найдены.\n");  
 }  
} while (searchChoice != 0);  
  
free(students);  
  
return 0;

}

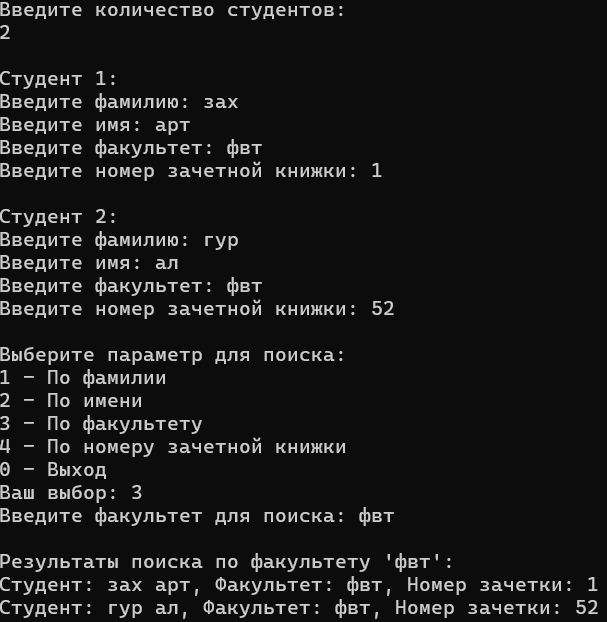
**Результат работы программ**

Результаты работы программы показаны на рисунке 1.



**Рисунок 1 — Результаты работы программы**

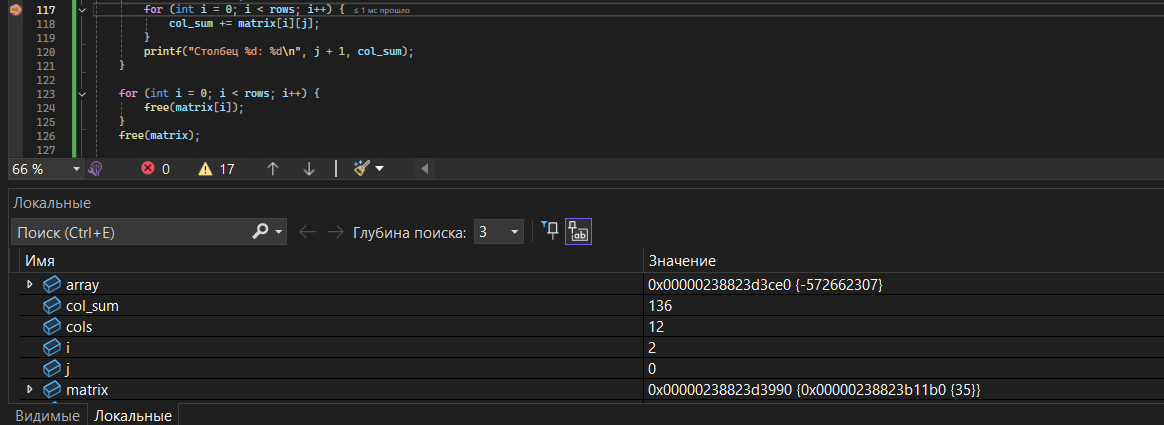
Результат второй работы программы показан на рисунке 2



**Рисунок 2 — Результаты работы программы**

**Протокол трассировки программы**

На рисунке 3 показан протокол трассировки после обработки значений переменных



**Рисунок 3 — Протокол трассировки**

**Вывод:**

В ходе выполнения лабораторной работы разработали программу, вычисляющую разницу между максимальным и минимальным элементами массива, реализующую инициализацию массива случайными числами, реализующую создание массива произвольного размера, вводимого с клавиатуры, вычисляющую сумму значений в каждом столбце (строке) двумерного массива, осуществляющую поиск среди структур student структуру с нужными параметрами (фамилией, именем, факультетом и номером зачётной книжки).

Задание: выполнили поиск по подстроке, добавили сохранение матриц, добавили ввод букв в графе зачетная книжка.