Министерство науки и высшего образования РФ

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1

по дисциплине «логика и основы алгоритмизации в инженерных задачах»

на тему «простые структуры данных»

Выполнили:

студенты группы 24ВВВ4

Королёв Д.В

Алешин К.А

Приняли:

Юрова О.В

Деев М.В

Пенза 2024

**Цель работы** – вспомнить базовые концепции языка и простые структуры данных.

**Общие сведения** – к простым структурам данных языка С относятся массивы, строки. Структуры это простейшая составная структура данных.

**Листинг**

**Задача 1**

#include <cstdio>

#include <cstdint>

#include <climits>

int main() {

uint32\_t arr[] = { 1,2,3,4,5,6,7,8,9,10,213,12355,6,21 };

#define UINT32\_MIN 0

uint32\_t max = UINT32\_MIN;

#undef UINT32\_MIN

uint32\_t min = UINT32\_MAX;

for (size\_t i = 0; i < sizeof(arr) / 4; ++i) {

if (min > arr[i]) { min = arr[i]; }

if (max < arr[i]) { max = arr[i]; }

}

printf("%d\n", max);

printf("%d\n", min);

return 0;

}

**Задача 2**

#include <cstdio>

#include <cstdint>

#include <climits>

#include <ctime>

#include <cstdlib>

int main() {

uint32\_t arr[20];

for (size\_t i = 0; i < sizeof(arr) / sizeof(uint32\_t); ++i) {

arr[i] = rand() % 100;

printf("%d ", arr[i]);

}

return 0;

}

**Задача 3**

#include <cstdio>

#include <cstdint>

#include <cstdlib>

#include <climits>

int main() {

uint32\_t size;

scanf("%d", &size);

uint32\_t\* ptr\_arr = (uint32\_t\*)malloc(size \* sizeof(uint32\_t));

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

scanf("%d", &ptr\_arr[i]);

}

printf("\n\n\n");

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

printf("%d\n", ptr\_arr[i]);

}

free(ptr\_arr);

return 0;

}

**Задача 4**

#include <cstdio>

#include <cstdint>

#include <climits>

#include <ctime>

#include <cstdlib>

int main() {

uint32\_t arr[10][10];

for (size\_t i = 0; i < 10; ++i) {

for (size\_t j = 0; j < 10; ++j) {

arr[i][j] = rand() % 250;

printf("%u\t", arr[i][j]);

}

printf("\n");

}

printf("\n\n\n");

uint32\_t row\_sum[10] = { 0 };

for (size\_t i = 0; i < 10; ++i) {

for (size\_t j = 0; j < 10; ++j)

row\_sum[i] += arr[i][j];

}

for (size\_t i = 0; i < 10; ++i) {

printf("%d\n", row\_sum[i]);

}

return 0;

}

**Задача 5**

#include <cstdint>

typedef unsigned long long size\_t;

extern "C" int scanf(const char\*, ...);

extern "C" int printf(const char\*, ...);

extern "C" void\* malloc(size\_t);

extern "C" void free(void\*);

extern "C" int memcmp(const void\*, const void\*, size\_t);

extern "C" size\_t strlen(const char\* str);

extern "C" int strcmp(const char\*, const char\*);

#define ERROR 1

typedef char surname\_student[100];

typedef char name\_student[100];

typedef uint8\_t course\_student;

struct table\_student {

name\_student name;

surname\_student surname;

course\_student course;

};

table\_student\* ptr\_to\_struct = nullptr;

int set\_student\_table(size\_t& SIZE) {

printf("Enter table size:\n");

scanf("%zu", &SIZE);

ptr\_to\_struct = (table\_student\*)malloc(sizeof(table\_student) \* SIZE);

if (ptr\_to\_struct == nullptr) {

printf("Memory allocation failed\n");

return ERROR;

}

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

printf("Enter name for student %zu:\n", i + 1);

scanf("%99s", ptr\_to\_struct[i].name);

printf("Enter surname for student %zu:\n", i + 1);

scanf("%99s", ptr\_to\_struct[i].surname);

printf("Enter course for student %zu:\n", i + 1);

scanf("%hhu", &ptr\_to\_struct[i].course);

}

return 0;

}

int find\_student(table\_student\* ptr\_to\_struct, size\_t SIZE, name\_student nm, surname\_student snm) {

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

if (strcmp(ptr\_to\_struct[i].name, nm) == 0 &&

strcmp(ptr\_to\_struct[i].surname, snm) == 0) {

printf("User found\n");

return 0;

}

}

return ERROR;

}

int main() {

size\_t SIZE = 0;

uint8\_t descriptor\_set = set\_student\_table(SIZE);

if (descriptor\_set != 0) {

printf("Error in set\_student\_table()\n");

return ERROR;

}

name\_student buffer\_name;

printf("Enter (name): \n");

scanf("%99s", buffer\_name);

surname\_student buffer\_surname;

printf("Enter (surname): \n");

scanf("%99s", buffer\_surname);

uint8\_t descriptor\_find = find\_student(ptr\_to\_struct, SIZE, buffer\_name, buffer\_surname);

if (descriptor\_find == ERROR) {

printf("Not found\n");

}

free(ptr\_to\_struct);

return 0;

}

Рисунок 1– Результат выполнения 1-го задания

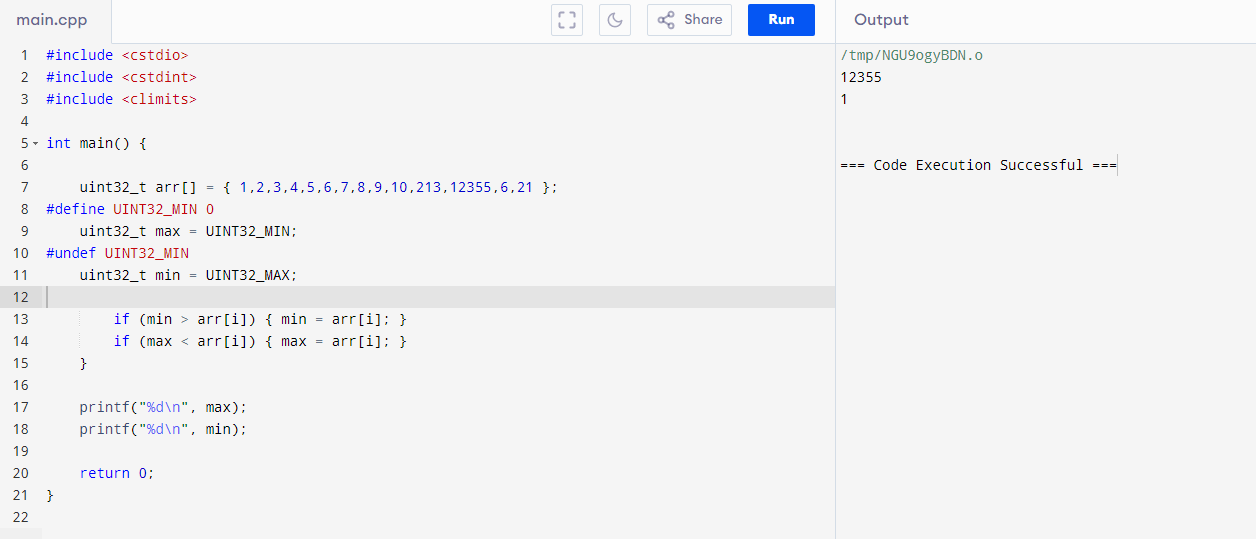


Рисунок 2– Результат выполнения 2-го задания

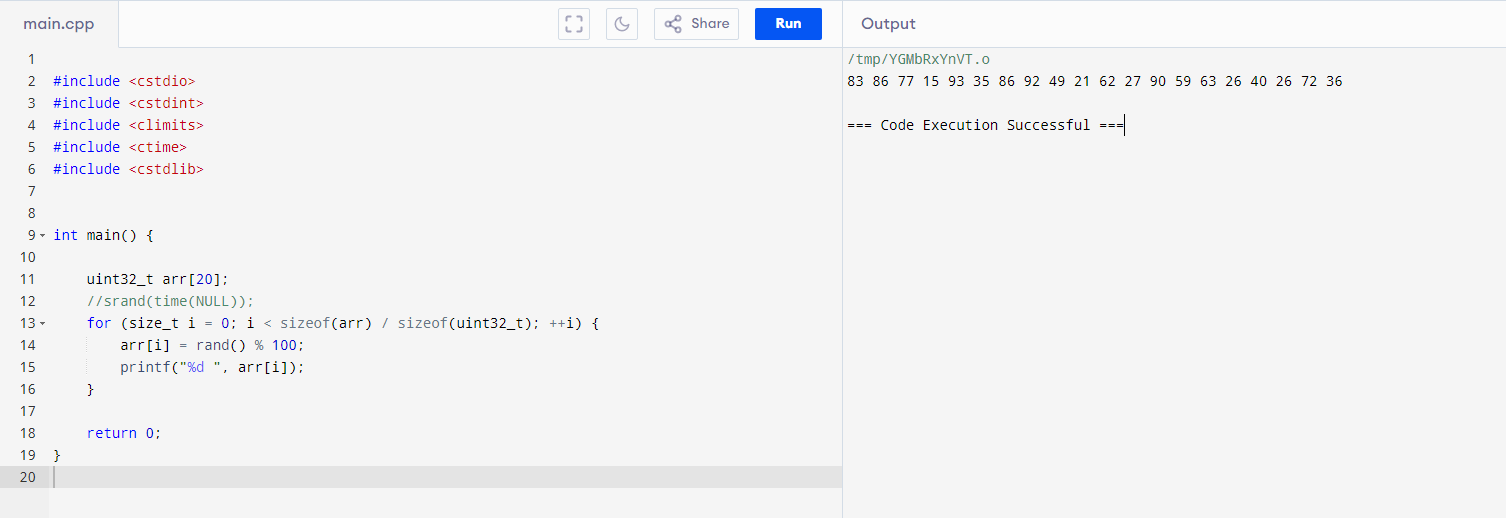


Рисунок 3– Результат выполнения 3-го задания

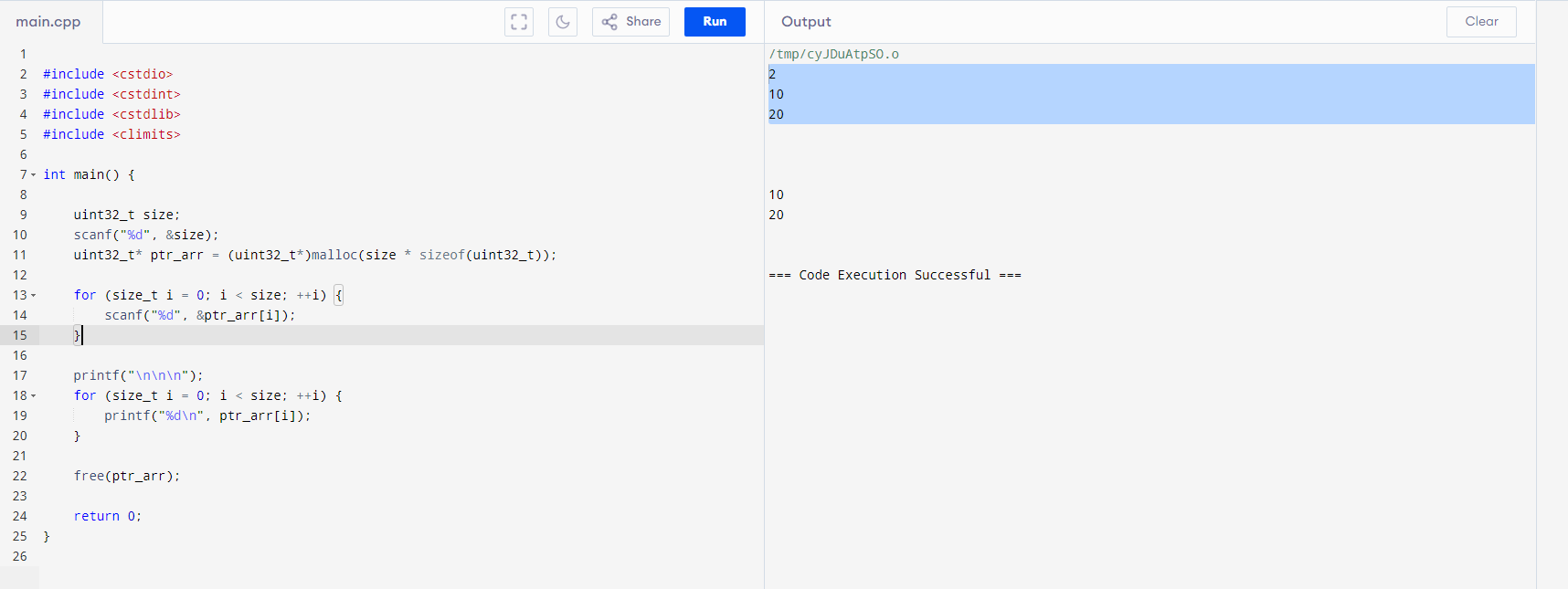


Рисунок 4 – Результат выполнения 4-го задания

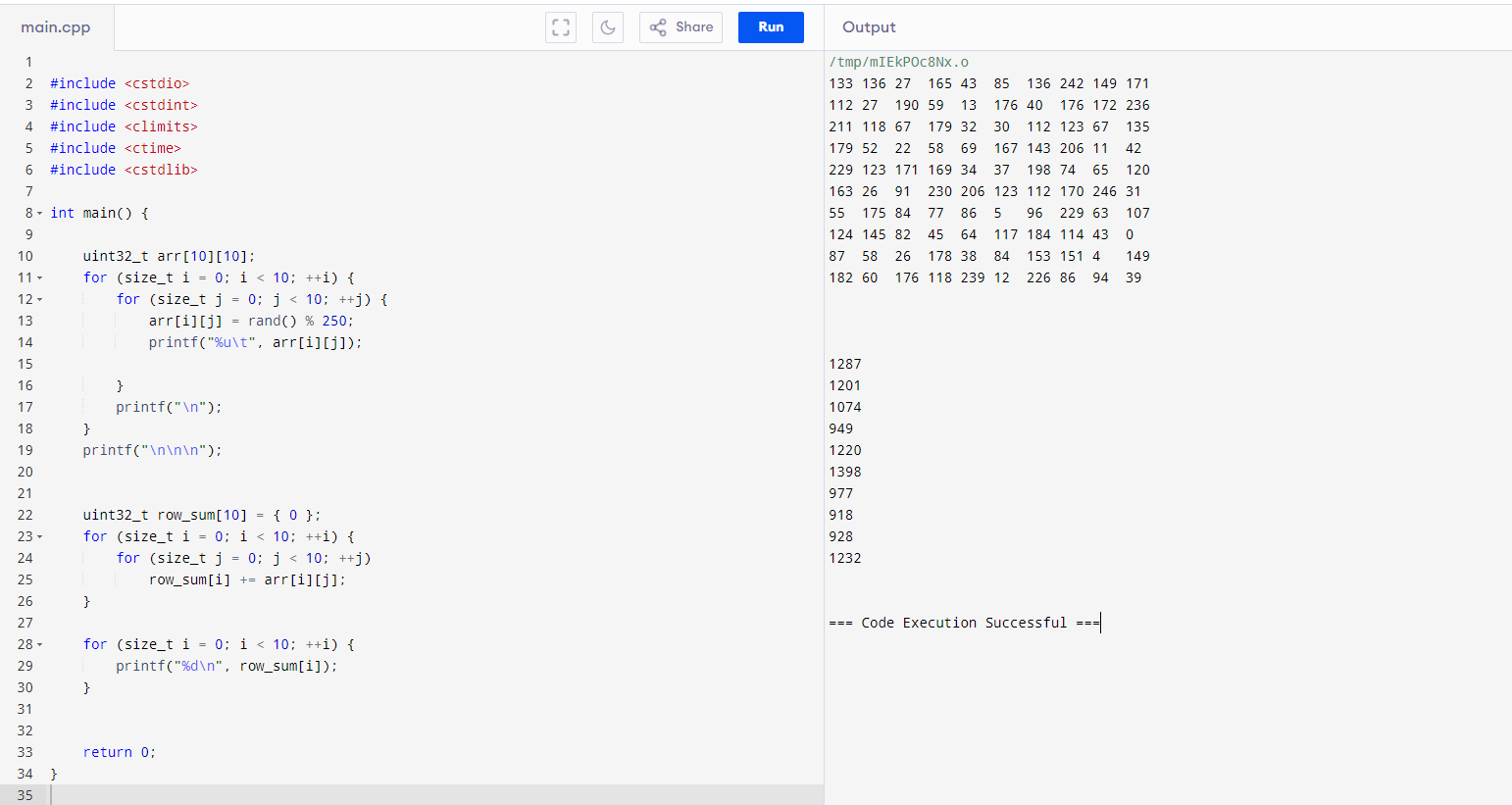
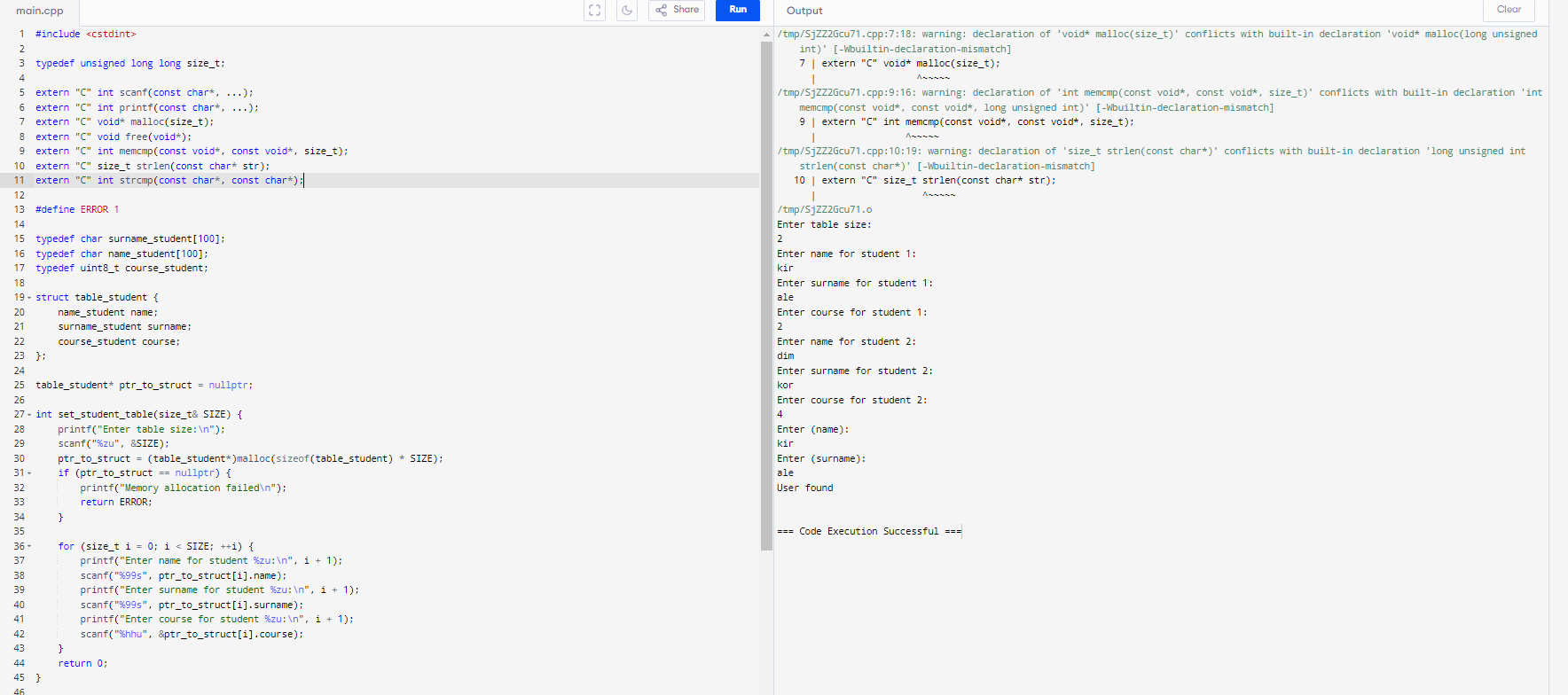


Рисунок 5 – Результат выполнения 5-го задания



**Вывод**: в ходе выполнения лабораторной работы были освежены навыки работы с базовыми структурами данных, строками и массивами.