

Министерство науки и образования Российской Федерации
Пензенский государственный университет
Кафедра «Вычислительная техника»

ОТЧЕТ

по лабораторной работе №1
по курсу «Логика и основы алгоритмизации
в инженерных задачах»
на тему «Простые структуры данных»

Выполнил:
студент группы 23ВВВ4
Костин К. А.

Приняли:
Демеев М. В.
Юрова О. В.

Пенза 2024

Общие сведения:

К простым структурам данных языка Си относятся массивы, строки (массивы символов). Структуры (struct) – простейшая составная структура данных.

Цель:

Научиться работать с простыми структурами данных

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

```
> ./a.out
Введите кол-во столбцов и через пробел кол-во строк > 3 3
32 74 6          112
13 83 8          104
61 97 -11        147

разница = 108
```

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

```
> ./a.out
Введите кол-во столбцов и через пробел кол-во строк > 3 3
32 74 6          112
13 83 8          104
61 97 -11        147

разница = 108
```

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

```
> ./a.out
Введите кол-во столбцов и через пробел кол-во строк > 3 3
32 74 6          112
13 83 8          104
61 97 -11        147

разница = 108
```

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

```
> ./a.out
Введите кол-во столбцов и через пробел кол-во строк > 3 3
32 74 6          112
13 83 8          104
61 97 -11        147

разница = 108
```

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

```
Введите имя студента
> Kirill1
Введите фамилию студента
> K1
Введите название факультета студента K1 Kirill1
> fvt
Введите номер зачётной книжки студента K1 Kirill1
> 123
Студент K1 Kirill1 обучается на факультете fvt, номер зачётной книжки 123
> Введите имя студента
> Kirill2
Введите фамилию студента
> K2
Введите название факультета студента K2 Kirill2
> fvt
Введите номер зачётной книжки студента K2 Kirill2
> 234
Студент K2 Kirill2 обучается на факультете fvt, номер зачётной книжки 234
> Введите имя студента
> *
по какому параметру ищем?
1 - имя
2 - фамилия
3 - название факультета
4 - номер зачётной книжки
> 1
Введите имя > Kirill1
Студент K1 Kirill1 обучается на факультете fvt, номер зачётной книжки 123
```

Листинг:

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <string.h>
```

```
struct student {

    char famil[20];
```

```

char name[20];

char facult[20];

int Nomzach;

struct student *next;

};

```

```

struct student *create_student(const char *name, const char *famil, const char *facult, int Nomzach);

void add_student(struct student **head, const char *name, const char *famil, const char *facult, int Nomzach);

void search_by_fname(const char *fname, struct student *head);

void search_by_sname(const char *sname, struct student *head);

void search_by_facultet(const char *facultet, struct student *head);

void search_by_number(int number, struct student *head);

void free_list(struct student *head);

```

```

int main(void) {

    srand(time(NULL));

    int size_mass_col, size_mass_row, MAX_VAL = -40, MIN_VAL = 100, sum = 0;


    printf("Введите кол-во столбцов и через пробел кол-во строк > ");

    scanf("%d %d", &size_mass_col, &size_mass_row);


    int **mass = (int **)malloc(size_mass_col * sizeof(int *));

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

        mass[i] = (int *)malloc(size_mass_row * sizeof(int));

    }


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

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

            mass[i][j] = rand() % 141 - 40;

            sum += mass[i][j];

            printf("%d ", mass[i][j]);

            if (mass[i][j] > MAX_VAL) { MAX_VAL = mass[i][j]; }

            if (mass[i][j] < MIN_VAL) { MIN_VAL = mass[i][j]; }

        }

    }
}

```

```

    }

    printf("\t %d\n", sum);

    sum = 0;
}

printf("\n\n");

printf("разница = %d\n", MAX_VAL - MIN_VAL);

struct student *head = NULL;
while (1) {
    char name[20], famil[20], facult[20];

    int Nomzach;

    printf("Введите имя студента\n> ");
    scanf("%s", name);
    if (strcmp(name, "") == 0) {
        break;
    }
    printf("Введите фамилию студента\n> ");
    scanf("%s", famil);
    printf("Введите название факультета студента %s %s\n> ", famil, name);
    scanf("%s", facult);
    printf("Введите номер зачётной книжки студента %s %s\n> ", famil, name);
    scanf("%d", &Nomzach);

    printf("Студент %s %s обучается на факультете %s, номер зачётной книжки %d\n> ", famil, name,
    facult, Nomzach);

    add_student(&head, name, famil, facult, Nomzach);
}

printf("по какому параметру ищем?\n1 - имя\n2 - фамилия\n3 - название факультета\n4 - номер
зачётной книжки\n> ");

int search;

scanf("%d", &search);

```

```
char name_stud[100];
char sname_stud[100];
char facultet[100];
int number;
switch (search) {
case 1: {
    printf("Введите имя >");
    scanf("%s", name_stud);
    search_by_sname(name_stud, head);
    break;
}
case 2: {
    printf("Введите фамилию >");
    scanf("%s", sname_stud);
    search_by_fname(sname_stud, head);
    break;
}
case 3: {
    printf("Введите название факультета >");
    scanf("%s", facultet);
    search_by_facultet(facultet, head);
    break;
}
case 4: {
    printf("Введите номер книжки >");
    scanf("%d", &number);
    search_by_number(number, head);
    break;
}
default:
    printf("неправильный ввод");
}
```

```

// Освобождение памяти
for (int i = 0; i < size_mass_col; i++) {
    free(mass[i]);
}
free(mass);

free_list(head);

return 0;
}

struct student *create_student(const char *name, const char *famil, const char *facult, int Nomzach) {
    struct student *new_student = (struct student *)malloc(sizeof(struct student));
    strcpy(new_student->name, name);
    strcpy(new_student->famil, famil);
    strcpy(new_student->facult, facult);
    new_student->Nomzach = Nomzach;
    new_student->next = NULL;
    return new_student;
}

void add_student(struct student **head, const char *name, const char *famil, const char *facult, int
Nomzach) {
    struct student *new_student = create_student(name, famil, facult, Nomzach);
    new_student->next = *head;
    *head = new_student;
}

void search_by_fname(const char *fname, struct student *head) {
    struct student *current = head;
    while (current != NULL) {
        if (strcmp(current->famil, fname) == 0) {

```

```

        printf("Студент %s %s обучается на факультете %s, номер зачётной книжки %d \n", current-
>famil, current->name, current->facult, current->Nomzach);

        return;

    }

    current = current->next;

}

printf("Студент %s не найден.\n", fname);
}

```

```

void search_by_sname(const char *sname, struct student *head) {

    struct student *current = head;

    while (current != NULL) {

        if (strcmp(current->name, sname) == 0) {

            printf("Студент %s %s обучается на факультете %s, номер зачётной книжки %d \n", current-
>famil, current->name, current->facult, current->Nomzach);

            return;

        }

        current = current->next;

    }

    printf("Студент %s не найден.\n", sname);

}

```

```

void search_by_facultet(const char *facultet, struct student *head) {

    struct student *current = head;

    while (current != NULL) {

        if (strcmp(current->facult, facultet) == 0) {

            printf("Студент %s %s обучается на факультете %s, номер зачётной книжки %d \n", current-
>famil, current->name, current->facult, current->Nomzach);

            return;

        }

        current = current->next;

    }

    printf("Студент не найден на факультете %s.\n", facultet);

}

```



```

void search_by_number(int number, struct student *head) {
    struct student *current = head;
    while (current != NULL) {
        if (current->Nomzach == number) {
            printf("Студент %s %s обучается на факультете %s, номер зачётной книжки %d \n", current-
>famil, current->name, current->facult, current->Nomzach);
            return;
        }
        current = current->next;
    }
    printf("Нет книжки с номером %d.\n", number);
}

```

```

void free_list(struct student *head) {
    struct student *current = head;
    struct student *next;
    while (current != NULL) {
        next = current->next;
        free(current);
        current = next;
    }
}

```

Вывод: в ходе выполнения этой лабораторной работы были получены навыки работы со структурами и массивами