

ГУАП

КАФЕДРА № 34

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

старший преподаватель

должность, уч. степень, звание

подпись, дата

К.А.Жиданов

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 2

по курсу: ЯЗЫКИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

3145

подпись, дата

Д.А.Смертин

инициалы, фамилия

Санкт-Петербург 2022

ЦЕЛЬ РАБОТЫ

Вариант 5 - построение графа.

ХОД РАБОТЫ

Данная программа по созданию графов реализована при помощи метода матрицы смежности.

	1	2	3	4
1	0	1	0	1
2	0	0	1	1
3	0	1	0	0
4	1	0	1	0

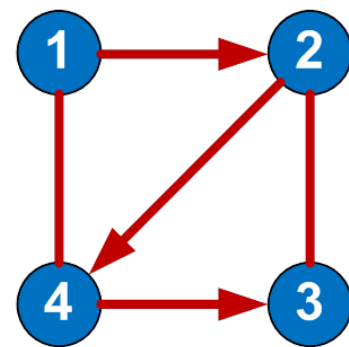


рис.1. Матрица смежности

Итак, моя программа встречает пользователя приветственным сообщением с предложением провзаимодействовать, на фотографии видно, каким образом. Можно создать граф, удалить файл с графом и подробно прочесть про программу

```
int main()
{
    printf("%s", "Welcome to the Graphs, what do u want to do?\n\nOpen existing graph - O\nCreate graph - C\nDelete graph - D\nAbout program - A\nSo....: ");
    char choose[1024];
    char O[1024] = "O";
    char C[1024] = "C";
    char D[1024] = "D";
    char A[1024] = "A";
    scanf("%s", &choose);
    if (strcmp (choose, O) == 0) {
        printf("\n");
        opening();
    }
    else if (strcmp (choose, C) == 0) {
        printf("\n");
        creator();
    }
    else if (strcmp (choose, D) == 0) {
        deletegraph();
    }
    else if (strcmp (choose, A) == 0) {
        about();
    }
    return 0;
}
```

рис.2.Приветственное меню

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <locale.h>
#include <string.h>
int about() {

    printf("%s", "\nWelcome to the graph editor! It`s a program, what allows creating, editing and deleting graphs. ");
    printf("%s", "It works using adjacency matrix technology. It means, that if you want to see your graph, you will see a matrix, ");
    printf("%s", "created with your instructions. In this program '0' means edge`s absence, while '1' means edge`s existence. ");
    printf("%s", "'8' means impossibility of creating edge because of 1 knot.\n\n");
    printf("%s", "Good luck, fella!\n\n");

}

int deletograph() {

    remove("C:\\Users\\dsmet\\Desktop\\a.txt");
    printf("\n%s", "Graph is deleted successfully!");

}

```

рис.3.О программе

Ниже показано, как работает удаление файла с графом.

```

int deletograph() {

    remove("C:\\Users\\dsmet\\Desktop\\a.txt");
    printf("\n%s", "Graph is deleted successfully!");
    return 0;

}

```

рис.4.Удаление файла с графом

Нижe продемонстрировано открытие файла с данными о графе. В файле a.txt в репозитории есть вид файла.

```
]int opening() {  
    char size[255];  
    char arr[255];  
    FILE *file;  
    file = fopen("C:\\Users\\dsmet\\Desktop\\a.txt", "r");  
    fgets(size, 255, file);  
  
    fgets(arr, 255, file);  
  
    int x = size[0] - '0';  
    int graf[3][3];  
    memset(graf, 0, sizeof(graf));  
  
    int y = arr[1] - '0';  
    graf[0][1] = y;  
    //printf("%d", y);  
    int j = 0;  
  
    for(int i = 0; i < x; i++)  
    {  
        for (int k = 0; k < x; k++)  
        {  
  
            y = arr[j] - '0';  
            graf[i][k] = y;  
            j = j + 1;  
        }  
  
    }  
  
    fclose(file);  
  
    edit(x, graf);  
-}
```

рис.5.Открытие файла с графом

Функция, которая создает граф. Позволяет выбрать количество узлов, а также обладает приветственным сообщением.

```
int creator()
{
    printf("Welcome to the graph creator!\n\nWhat's the count of knots would you like to create?\nN = ");
    int x;
    scanf("%d", &x);
    int graf[x][x];
    int i, k;
    for(i = 0; i < x; i++)
    {
        for(k = 0; k < x; k++)
        {
            graf[i][k] = 0;
        }
    }
    printf("\n");
    printf("%s %d %s", "Graph with", x, "edges has created successfully!\n\nWould you like to edit your graph?\n");
    if (answer() == 1){
        printf("\n");
        edit(x, graf);
    }
    else
        printf("%s", "Okay, bye!");
    return graf;
}
```

рис.6. Функция создания графа

Функция, в которую переходит программа при создании графа. Позволяет создать/удалить ребра и узлы, увидеть, как выглядит граф на данный момент, а также перейти в загрузку графа в файл.

```
int edit(int x, int graf[][x]){
    char AE[1024] = "AE";
    char DE[1024] = "DE";
    char AK[1024] = "AK";
    char DK[1024] = "DK";
    char S[1024] = "S";
    char E[1024] = "E";
    char choose[1024];

    printf("%s", "What do you want to do with your graph?\n\nAdd edges - AE\nDelete edges - DE\nAdd knots - AK\nDelete knots - DK\nSee my graph - S\nExit - E\nSo...:");
    scanf("%s", &choose);

    if (strcmp (choose, AE) == 0){
        rebrocreator(x, graf);
        edit(x, graf);
    }
    else if (strcmp (choose, DE) == 0){
        rebrodeleter(x, graf);
        edit(x, graf);
    }
    else if (strcmp (choose, AK) == 0){
        adder(x, graf);
        edit(x, graf);
    }
    else if (strcmp (choose, DK) == 0){
        deleter(x, graf);
        edit(x, graf);
    }
    else if (strcmp (choose, S) == 0){
        observe(x, graf);
        edit(x, graf);
    }
    if (strcmp (choose, E) == 0){
        upload(x, graf);
    }
}
```

рис.7. Функция редактирования графа

```

What do you want to do with your graph?
Add edges - AE
Delete edges - DE
Add knots - AK
Delete knots - DK
See my graph - S
Exit - E

So...:S

| 1 2 3
1 X 1 0
2 1 X 0
3 0 0 X

```

рис.8. Вывод графа в консоль

Созданием ребер занимается нижеуказанная функция.

```

int rebrocreater(int x, int graf[][x]){
    int i, k;
    printf("%s", "Which knots do you want to connect?\nFirst: ");

    scanf("%d", &i);

    if (i > x || i < 0)
    {
        printf("%s", "Choose another!\n");
        rebrocreater(x, graf);
    }
    printf("%s", "And second: ");
    scanf("%d", &k);
    if (k > x || k < 0)
    {
        printf("%s", "Choose another!\n");
        rebrocreater(x, graf);
    }
    if (i != k){
        graf[i - 1][k - 1] = 1;
        graf[k - 1][i - 1] = 1;
        printf("%s", "Edge is created successfully!\n\n");

        return graf;
    }
    else
        printf("%s", "Error! choose another knots!\n\n");
        rebrocreater(x, graf);
}

```

рис.9.Создание ребер

Ниже указана функция, удаляющая ребра

```
int rebrodeleter(int x, int graf[][x]){
    int i, k;
    printf("%s", "Which knots do you want to disconnect?\nFirst: ");

    scanf("%d", &i);

    if (i > x || i < 0)
    {
        printf("%s", "Choose another!\n");
        rebrocreator(x, graf);
    }
    printf("%s", "And second: ");
    scanf("%d", &k);
    if (k > x || k < 0)
    {
        printf("%s", "Choose another!\n");
        rebrocreator(x, graf);
    }
    if (i != k){
        graf[i - 1][k - 1] = 0;
        graf[k - 1][i - 1] = 0;
        printf("%s", "Edge is deleted successfully!\n\n");

        return graf;
    }
    else
        printf("%s", "Error! choose another knots!\n\n");
        rebrocreator(x, graf);
}
```

рис.10. Удаление ребер

Ниже указана функция, добавляющая узлы

```

int adder(int x, int graf[][x]){
    printf("%s", "How many knots do you want to add?\n\nN = ");
    int a;
    scanf("%d", &a);
    int c = x + a;
    int pgraf[c][c];
    int i, k;
    memset(pgraf, 0, sizeof(pgraf));

    for(i = 0; i < x; i++)
    {
        for(k = 0; k < x; k++)
        {
            pgraf[i][k] = graf[i][k];

        }
    }

    x = c;
    printf("%s", "Knots are added successfully!\n\n");
    edit(x, pgraf);
}

```

рис.11. Добавление узлов

Далее функция удаления узлов

```

int deleter(int x, int graf[][x])
{
    printf("%s", "How many knots do you want to delete?\n\nN = ");
    int a;
    scanf("%d", &a);
    int c = x - a;
    int pgraf[c][c];
    int i, k;
    memset(pgraf, 0, sizeof(pgraf));

    for(i = 0; i < x; i++)
    {
        for(k = 0; k < x; k++)
        {
            pgraf[i][k] = graf[i][k];

        }
    }

    x = c;
    printf("%s", "Knots are deleted successfully!\n\n");
    edit(x, pgraf);
}

```


Потом функция, которая позволяет увидеть наш граф

```
int observe(int x, int graf[][x])
{
    printf("\n");
    int i, k;
    int c = x;
    int igraf[c][c];
    char sgraf[c][c];

    printf("%c ", '|');

    for (i = 0; i < c; i++)
    {
        for (k = 0; k < c; k++)
        {
            if (i == k)
                sgraf[i][k] = 'X';
            else if (graf[i][k] == 0)
                sgraf[i][k] = '0';
            else if (graf[i][k] == 1)
                sgraf[i][k] = '1';
        }
    }

    sgraf[0][0] = 'X';
    for (i = 0; i < c; i++)
    {
        igraf[i][0] = i + 1;
        igraf[0][i] = i + 1;
        printf("%d ", igraf[i][0]);

    }
    printf("\n");
    for (i = 0; i < c; i++)
    {
        printf("%d ", igraf[0][i]);
        for (k = 0; k < c; k++)
        {
            if (k >= 9)
                printf("%c ", sgraf[i][k]);
            else
                printf("%c ", sgraf[i][k]);
        }
        printf("\n");
    }
    printf("\n");
}
```

рис.12.1.Вывод графа в консоль

```

What do you want to do with your graph?
E
Add edges - AE
Delete edges - DE
Add knots - AK
Delete knots - DK
See my graph - S
Exit - E

So...:S

| 1 2 3
1 X 1 0
2 1 X 0
3 0 0 X

```

рис.12.2. Вывод графа в консоль

Далее указаны использованные библиотеки и инструкция

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <locale.h>
#include <string.h>
int about(){
    printf("%s", "\nWelcome to the graph editor! It's a program, what allows creating, editing and deleting graphs. ");
    printf("%s", "It works using adjacency matrix technology. It means, that if you want to see your graph, you will see a matrix, ");
    printf("%s", "created with your instructions. In this program '0' means edge's absence, while '1' means edge's existence. ");
    printf("%s", "'X' means impossibility of creating edge because of 1 knot.\n\n");
    printf("%s", "Good luck, fella!\n\n");
    return 0;
}

```

рис.13.Библиотеки и FAQ :)

ВЫВОДЫ

Написал программу для создания графа, управления им.

