№4.

#include <stdio.h>

#include <stdlib.h>

int\* bubble\_sort(int\* arr, int size) {

    int\* sorted\_array = malloc(size \* sizeof(int));

    for (int i = 0; i < size; i++) {

        sorted\_array[i] = arr[i];

    }

    for (int i = 0; i < size - 1; i++) {

        for (int j = 0; j < size - i - 1; j++) {

            if (sorted\_array[j] > sorted\_array[j + 1]) {

                int temp = sorted\_array[j];

                sorted\_array[j] = sorted\_array[j + 1];

                sorted\_array[j + 1] = temp;

            }

        }

    }

    return sorted\_array;

}

int main() {

    int n = 0;

    printf("Enter array's size: ");

    scanf("%d", &n);

    int\* a = malloc(n \* sizeof(int));

    for (int i = 0; i < n; i++) {

        printf("Enter %d array's number: ", i + 1);

        scanf("%d", &a[i]);

    }

    int\* sorted\_array = bubble\_sort(a, n);

    printf("Sorted array: ");

    for (int i = 0; i < n; i++) {

        printf("%d ", sorted\_array[i]);

    }

    printf("\n");

    free(a);

    free(sorted\_array);

    return 0;

}

№5.

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <windows.h>

double opred(double\*\* b, int size) {

    int swaps = 0;

    for (int i = 0; i < size; i++) {

        if (b[i][i] == 0) {

            for (int j = i + 1; j < size; j++) {

                if (b[j][i] != 0) {

                    double\* temp = b[i];

                    b[i] = b[j];

                    b[j] = temp;

                    swaps++;

                    break;

                }

            }

            if (b[i][i] == 0) {

                return 0;

            }

        }

        for (int j = i + 1; j < size; j++) {

            double mn = b[j][i] / b[i][i];

            for (int k = i; k < size; k++) {

                b[j][k] -= mn \* b[i][k];

            }

        }

        printf("\nМатрица после %d-го шага:\n", i + 1);

        for (int x = 0; x < size; x++) {

            for (int y = 0; y < size; y++) {

                printf("%.2f ", b[x][y]);

            }

            printf("\n");

        }

        printf("\n");

    }

    double determinant = 1;

    for (int i = 0; i < size; i++) {

        determinant \*= b[i][i];

    }

    if (swaps % 2 != 0) {

        determinant = -determinant;

    }

    return determinant;

}

int main() {

    SetConsoleCP(CP\_UTF8);

    SetConsoleOutputCP(CP\_UTF8);

    int choose;

    printf("Введите 1 если вы хотите автоматически заполнить матрицу, введите 2 если вручную: ");

    scanf("%d", &choose);

    if (choose != 1 && choose != 2) {

        printf("Ошибка: введите 1 или 2.\n");

        return 1;

    }

    int n;

    printf("Введите размер: ");

    scanf("%d", &n);

    double\*\* b = malloc(sizeof(double\*) \* n);

    for (int i = 0; i < n; i++) {

        b[i] = malloc(sizeof(double) \* n);

    }

    switch (choose) {

        case 1:

            srand(time(NULL));

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

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

                    b[i][j] = rand() % 10;

                }

            }

            printf("Случайный массив:\n");

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

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

                    printf("%.2f ", b[i][j]);

                }

                printf("\n");

            }

            break;

        case 2:

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

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

                    printf("Введите %d элемент %d строки: ", j + 1, i + 1);

                    scanf("%lf", &b[i][j]);

                }

            }

            break;

    }

    double determinant = opred(b, n);

    if (determinant == 0) {

        printf("\nОпределитель равен нулю.\n");

    } else {

        printf("Определитель = %.2f\n", determinant);

    }

    for (int i = 0; i < n; i++) {

        free(b[i]);

    }

    free(b);

    return 0;

}

№6.

#include <stdio.h>

#include <string.h>

#include <windows.h>

#include <dirent.h>

void del();

void enter();

void list();

void edit\_a();

void reg();

int main() {

    SetConsoleCP(CP\_UTF8);

    SetConsoleOutputCP(CP\_UTF8);

    int choose;

    while (1) {

        printf("\n\n\nЕсли Вы хотите войти в учетную запись, введите 1.\nЕсли хотите добавить нового пользователя, введите 2.\nЧтобы увидеть список пользователей введите 3.\nЧтобы удалить пользователя введите 4.\nЧтобы изменить пароль пользователя введите 5.\nДля выхода введите 0: ");

        scanf("%d", &choose);

        switch (choose) {

            case 1:

                enter();

                break;

            case 2:

                reg();

                break;

            case 3:

                list();

                break;

            case 4:

                del();

                break;

            case 5:

                edit\_a();

                break;

            case 0:

                printf("Выход из программы.\n");

                return 0;

            default:

                printf("Вы ввели некорректное значение.\n");

        }

    }

    return 0;

}

void del(){

    printf("Введите логин пользователя, которого нужно удалить: ");

    char login[25];

    scanf("%s",login);

    char path[50];

    sprintf(path,"6/%s",login);

    FILE\* fp = fopen(path, "r");

    if (fp == NULL) {

        printf("Пользователь не найден.");

        return;

    }

    printf("Введите пароль от логина %s: ",login);

    char pass[50];

    scanf("%s",pass);

    char buffer[50];

    fgets(buffer,50,fp);

    //printf("%s %s",pass,buffer);

    //buffer[strcspn(buffer, "\n")] = 0;

    if (strcmp(buffer, pass) == 0){

        fclose(fp);

        remove(path);

        printf("Пользователь %s удалён",login);

    }

    else {

        printf("Пароль неправильный, пользователь не удалён");

        fclose(fp);

    }

}

void edit\_a() {

        struct dirent \*dp;

        DIR \*dir = opendir("6/");

        if (!dir) return;

        while ((dp = readdir(dir)) != NULL) {

            if (strcmp(dp->d\_name, ".") != 0 && strcmp(dp->d\_name, "..") != 0) {

                printf("%s\n", dp->d\_name);

            }

        }

        closedir(dir);

        printf("Введите логин для которого нужно изменить пароль: ");

        char login[25];

        scanf("%s", login);

        char path[50];

        sprintf(path, "6/%s", login);

        FILE\* fp = fopen(path, "w");

        if (fp == NULL) {

            printf("Пользователь не найден.\n");

            return;

        }

        printf("Введите новый пароль для пользователя %s: ", login);

        char new\_password[25];

        scanf("%s", new\_password);

        fprintf(fp, "%s", new\_password);

        fclose(fp);

        printf("Пароль для пользователя %s успешно изменен.\n", login);

}

void list(){

        struct dirent \*dp;

        DIR \*dir = opendir("6/");

        if (!dir) return;

        while ((dp = readdir(dir)) != NULL) {

            if (strcmp(dp->d\_name, ".") != 0 && strcmp(dp->d\_name, "..") != 0) {

                printf("%s\n", dp->d\_name);

    }

}

 closedir(dir);

}

void reg() {

        printf("Введите логин нового пользователя: ");

        char login[25];

        scanf("%s",login);

        //login[strcspn(login, "\n")] = 0;

        char path[50];

        sprintf(path,"6/%s",login);

        FILE\* fp = fopen(path,"a+");

        printf("\nВведите пароль для пользователя %s: ",login);

        char pass[50];

        scanf("%s",pass);

        fprintf(fp,"%s",pass);

        printf("Пользователь %s с паролем %s создан.",login,pass);

        fclose(fp);

}

void enter() {

    printf("Введите логин: ");

    char login[25];

    scanf("%s",login);

    char path[50];

    sprintf(path,"6/%s",login);

    FILE\* fp = fopen(path, "r");

    if (fp == NULL) {

        printf("Пользователь не найден.");

        return;

    }

    printf("Введите пароль от логина %s: ",login);

    char pass[50];

    scanf("%s",pass);

    char buffer[50];

    fgets(buffer,50,fp);

    //printf("%s %s",pass,buffer);

    //buffer[strcspn(buffer, "\n")] = 0;

    if (strcmp(buffer, pass) == 0){

        printf("Пароль верный");

    }

    else {

        printf("Пароль неправильный");

    }

    fclose(fp);

}