DCP5101: PROGRAM DESIGN

LECTURER: FARHAH AMALIYA BINTI ZAHARUDDIN

Group members:

NAME: MD FAHIM SHAHRIAR

ID: 1211205362

NAME: Shad Md Rakib Bin

ID: 1211206291

PROGRAM TYPE: MANAGEMENT

Title: Employee Management System Report

ASSIGNMENT GROUP: 22

INTRODUCTION

Title: Employee Management System Report

Introduction: The Employee Management System is a program designed to handle various tasks related to employee management in a company. It provides functionalities such as adding employees, maintaining an employee list, recording attendance, retrieving employee details, displaying company notices, and more. This report will provide an object of the program, its key features, and areas of improvement.

Objective of the program

The objective of the program is to create a management system for employee details in a company. The program allows users to perform various operations such as adding employees, displaying employee lists, marking attendance, viewing employee details, and publishing company notices. It utilizes a struct data type to store employee information such as name, ID, nationality, gender, age, rating, salary, and attendance. The program also includes functions for calculating salary tax based on employee salary and retrieving the company registration number. Overall, the program aims to provide a comprehensive solution for managing employee data and company-related tasks.

Key Features:

1.Adding Employees:

The program allows the user to add employee details such as name, nationality, gender, age, rating, and salary.

Each employee is assigned a unique ID automatically.

The entered details are stored in an array of structures.

2.Employee List:

The program displays a list of all employees, including their details, such as name, ID, nationality, gender, age, rating, and salary.

The list is generated by iterating over the array of structures.

3.Attendance:

The program enables the user to mark attendance for each employee.

The user can indicate whether an employee is present or absent by entering 'P' or any other character, respectively.

The program maintains a count of the number of days each employee has been present.

4.Employee Details:

The program allows the user to search for an employee by entering their ID.

Upon finding the employee, their complete details, including name, ID, nationality, gender, age, rating, salary, and the number of days present, are displayed.

Additionally, the program calculates and displays the salary after deducting a 6% tax.

5.Notice Board:

The program reads and displays the content of a file called "notice.txt" that contains company notices.

This feature allows the company to publish important information for employees.

6.Company Registration Number:

The program includes a function called companyRegNO that accepts a pointer to a character array.

This function is used to display the company's registration number.

The registration number is stored in a string variable arr.

When the user selects option 6, the program calls the companyRegNO function, passing the arr variable as an argument.

The function then prints the company's registration number to the console.

7.Program Termination:

When the user selects option 7, the program requests a password entry to confirm the termination.

The user is prompted to enter a password, and the input is stored in the variable run.

If the entered password matches the predefined password (123 in this case), the program terminates.

The value of ‘pass’ is set to ‘run’ to exit the main loop and end the program.

These two options provide additional functionalities to the program, allowing the user to view the company's registration number and terminate the program securely using a password.

Areas of Improvement:

1.Input Validation:

The program does not perform sufficient input validation, which can lead to unexpected behavior or crashes if invalid inputs are entered.

It is recommended to validate user inputs and handle errors gracefully to enhance the program's robustness.

2.Error Handling:

The program lacks comprehensive error handling. It does not provide clear error messages or handle exceptional cases gracefully.

It is advisable to implement appropriate error handling techniques, such as displaying meaningful error messages and handling file-related errors.

3.File Writing Capability:

The program currently only supports reading from the "notice.txt" file. It would be beneficial to extend its functionality to allow writing new notices to the file.

Program Design:

Input:

The program takes user input through the console.

Users can select options from a menu by entering corresponding numbers or characters.

Input is required for operations such as adding employees, marking attendance, and accessing employee details.

Process:

The program uses a struct data type to store employee details.

It maintains an array of employee structures to store multiple employee records.

The program uses loops, conditionals, and functions to perform different operations based on user input.

Functions are implemented to handle specific tasks such as adding employees, displaying employee lists, calculating salary tax, and publishing notices.

The program reads and writes data to a file to store and retrieve company notices.

Output:

The program displays information to the console based on the user's selection.

It outputs employee details, attendance records, company notices, and other relevant information.

Output is presented in a formatted manner to improve readability and user experience.

Conditions/Assumptions/Constraints:

The program assumes that the user will provide valid input when prompted.

It assumes a maximum limit of 50 employees in the system, as defined by the emp array.

The program assumes that the file "notice.txt" exists and can be accessed for reading company notices.

It assumes that the user has appropriate permissions to read and write files if required.

Error Handling:

The program checks for errors while opening the file for reading company notices.

If an error occurs, it displays an appropriate error message.

Error handling can be extended to handle other potential errors, such as invalid input, exceeding array bounds, or file write errors.

The program can include error messages and prompts to guide users in case of incorrect input or exceptional situations.

It can implement defensive programming techniques to handle unexpected scenarios and prevent crashes or undesired behavior.

Error handling can also include validating user input to ensure it meets specific criteria or constraints before processing the data.

Requirements implemented.

a) Arithmetic expression

b) Control structure

Selection

- if…else

```
if(employeeSize<0){
    printf("Empty List");
}
else{
        printf("\n--------------------Employee*List--------------------");
int i=0;
while(i<=employeeSize){
        printf("\nName: %s\nID: %d\nNatinality: %s\nSex: %s\nAge: %d\nRating: %.2f\nMonthly Salary %.2f\n\n",emp[i].name,emp[i].id,emp[i].nationality,emp[i].gender,emp[i].age,emp[i].rating,emp[i].salary);

    i++;
}
}
```

```
switch(functionalities){ // this switch case we used to use our all function

    case '1':
        employeeSize=employeeSize+1;
        add_employe();
        break;

    case '2':
        employeeList();
        break;

    case '3':
        attendance();

        break;

    case '4':
        printf("Enter Employee ID : ");
        int employeeID=1001;
        scanf("%d",&employeeID);
        employeedetails(employeeID);
        break;

    case '5':

        noticeFile();


        break;

    case '6':
        companyRegNO(arr);
        break;

    case '7':
        printf("\nEnter Password: ");
        scanf("%d",&run);
        printf("\n%d",run);
        break;

    default:
        printf("Invalid Choice");

        break;
}
```

Repetition

- do…while

```
//we used do while to print our switch case fucntion
do{

printf("\n\n\nPress 1.Add Employee : ");
printf("\nPress 2.Employee List: ");
printf("\nPress 3.Employee Attendance: ");
printf("\nPress 4.For employee full details: ");
printf("\nPress 5.Notice Board: ");
printf("\nPress 6.For watching company reg no: ");
printf("\nPress 7.For Ending: \n");

scanf(" %c",&functionalities);

switch(functionalities){ // this switch case we used to use our all function

    case '1':
        employeeSize=employeeSize+1;
        add_employe();
        break;

    case '2':
        employeeList();
        break;

    case '3':
        attendance();

        break;

    case '4':
        printf("Enter Employee ID : ");
        int employeeID=1001;
        scanf("%d",&employeeID);
        employeedetails(employeeID);
        break;

    case '5':

        noticeFile();
```

```
            break;

        case '6':
            companyRegNO(arr);
            break;

        case '7':
            printf("\nEnter Password: ");
            scanf("%d",&run);
            printf("\n%d",run);
            break;

        default:
            printf("Invalid Choice");

            break;
    }

    }
    while(pass!=run);
```

- for

```
for(attandanceCounter=0;attandanceCounter<=employeeSize;attandanceCounter++)
{
    printf("(%d) Name: %s\tID: %d For Present press P for Absent press any ",attandanceCounter+1,emp[attandanceCounter].name,emp[attandanceCounter].id);
    scanf(" %c",&presentVar);


    if(tolower(presentVar)!='p'){
        continue;

    }

    emp[attandanceCounter].present+=1;

}
```

- while

```
if(employeeSize<0){
    printf("Empty List");
}
else{
        printf("\n--------------------Employee*List--------------------");
int i=0;
while(i<=employeeSize){
        printf("\nName: %s\nID: %d\nNatinality: %s\nSex: %s\nAge: %d\nRating: %.2f\nMonthly Salary %.2f\n\n",emp[i].name,emp[i].id,emp[i].nationality,emp[i].gender,emp[i].age,emp[i].rating,emp[i].salary);

    i++;
}
}
```

- Array

```
char arr[] = {"\nReg.No: Fahim 1211205362"}; // this string we use as a our company reg number. we pass this string to function through pointer
```

- Pointers

```
void companyRegNO(char *arr) {  //this function we use to show company reg number. we had use here pointer type array
    printf("%s",arr);
}
```

```
//this is struct data type for build employee details array
    struct employeDetails{
    char name[30];
    int id;
    char nationality[20];
    char gender[7];
    int age;
    float rating;
    float salary;
    int present;
    }emp[50];
```

- File

```c
FILE *file;
char ch;

// Open the file in "r" (read) mode
file = fopen("notice.txt", "r");


if (file == NULL) {
    printf("Error opening the file.\n");
    return 1;
}

// Read and print each character until the end of the file is reached
while ((ch = fgetc(file)) != EOF) {
    printf("%c", ch);
}

// Close the file
fclose(file);

getch();
```

- Structure

```c
//this is struct data type for build employee details array
    struct employeDetails{
    char name[30];
    int id;
    char nationality[20];
    char gender[7];
    int age;
    float rating;
    float salary;
    int present;
    }emp[50];
```

User Manual

Adding an Employee:

Select option 1 from the menu.

Follow the prompts to enter the employee's details, including name, nationality, gender, age, rating, and salary.

The program will automatically assign an ID to the employee.

Once all details are entered, the program will display the employee's information.

Displaying Employee List:

Select option 2 from the menu.

The program will display a list of all employees and their details.

Marking Employee Attendance:

Select option 3 from the menu.

Follow the prompts to mark the attendance of each employee.

Enter 'P' for present or any other character for absent.

Viewing Employee Details:

Select option 4 from the menu.

Enter the employee's ID to view their full details, including name, nationality, gender, age, rating, salary, and attendance.

Viewing Company Notices:

Select option 5 from the menu.

The program will display any company notices that have been published in a file named "notice.txt".

Company Registration Number:

Select option 6 from the menu.

The program will display the company's registration number.

Exiting the Program:

To exit the program, select option 7 from the menu.

Source code with comments

```c
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

//this is struct data type for build employee details array
    struct employeDetails{
    char name[30];
    int id;
    char nationality[20];
    char gender[7];
    int age;
    float rating;
    float salary;
    int present;
    }emp[50];

    int employeeSize=-1; // this data type we take for employeeID it will take employee ID automatic


void add_employe(); // this function add for employee
void employeeList(); //this function for watching employee list
void attendance(); // this function for taking employee attendance
void employeedetails(employeeID); //this function we can search employee through their employee ID
int saleryTaxcalculate(employeeID); //this function will count employee tax base on their salary
void noticeFile(); //this function for company notice. we can publish notice though our file

//company reg number function with ptr
void companyRegNO(char *arr) {  //this function we use to show company reg number. we had use here pointer type array
    printf("%s",arr);
}


// this is main function
int main()
{
    int pass=123; //to run do loop. after use this pass our program will be off. until using this pass our program will be run
    int run=0;
    char functionalities; // this is for switch case function

    //this array use for pointer
    char arr[] = {"\nReg.No: Fahim 1211205342"}; // this string we use as a our company reg number. we pass this string to function through pointer




    printf("----------------------------Cyber*IT*Farm----------------------------\n"); // it's our company name
```

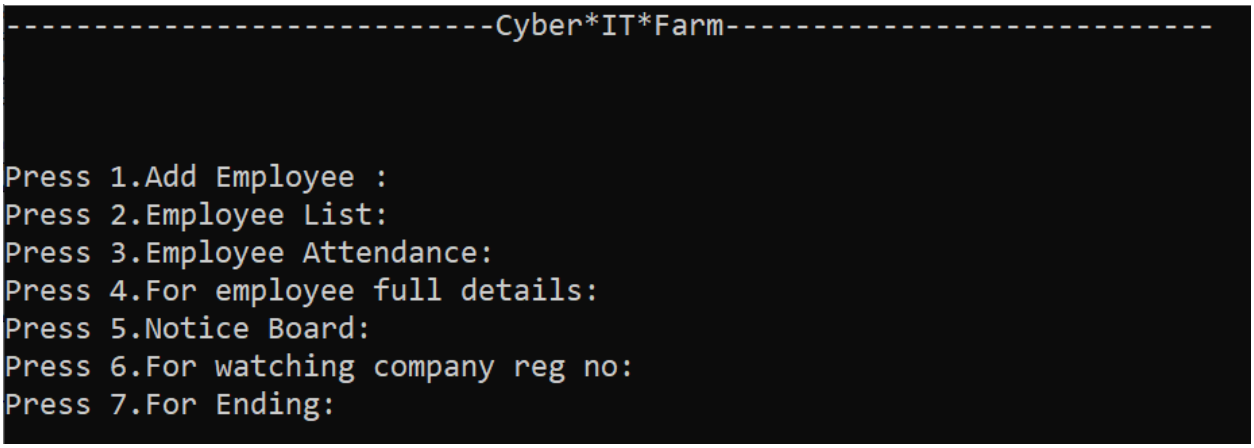
```c
    //we used do while to print our switch case fucntion
    do{

    printf("\n\n\nPress 1.Add Employee : ");
    printf("\nPress 2.Employee List: ");
    printf("\nPress 3.Employee Attendance: ");
    printf("\nPress 4.For employee full details: ");
    printf("\nPress 5.Notice Board: ");
    printf("\nPress 6.For watching company reg no: ");
    printf("\nPress 7.For Ending: \n");

    scanf(" %c",&functionalities);

    switch(functionalities){ // this switch case we used to use our all function

        case '1':
            employeeSize=employeeSize+1;
            add_employe();
            break;

        case '2':
            employeeList();
            break;

        case '3':
            attendance();

            break;

        case '4':
            printf("Enter Employee ID : ");
            int employeeID=1001;
            scanf("%d",&employeeID);
            employeedetails(employeeID);
            break;

        case '5':

            noticeFile();


            break;

        case '6':
            companyRegNO(arr);
            break;

        case '7':
            printf("\nEnter Password: ");
            scanf("%d",&run);
            printf("\n%d",run);
            break;

```

```
        default:
            printf("Invalid Choice");

            break;
    }

    }
    while(pass!=run);

}
 void add_employe()
{
    //create new employee

    printf("-----------------------------------------------Employe Details-----------------------------------------------\n");

    emp[employeeSize].id=1001+employeeSize; //Employee Id auto generate

    printf("Enter Name: ");
    scanf("%s",emp[employeeSize].name);

    printf("Enter nationality: ");
    scanf("%s",emp[employeeSize].nationality);

    printf("Enter Gender: ");
    scanf("%s",emp[employeeSize].gender);

    printf("Enter your age: ");
    scanf("%d",&emp[employeeSize].age);

    printf("Enter Rating: ");
    scanf("%f",&emp[employeeSize].rating);

    printf("Enter Salary: ");
    scanf("%f",&emp[employeeSize].salary);

    printf("\n\nName: %s\nID: %d\nNatinality: %s\nSex: %s\nAge: %d\nRating: %.2f\nMonthly Salary %.2f\n\n",emp[employeeSize].name,emp[employeeSize].id,emp[employeeSize].nationality,
        emp[employeeSize].gender,emp[employeeSize].age,emp[employeeSize].rating,emp[employeeSize].salary);

}
 void employeeList()
{
    if(employeeSize<0){
        printf("Empty List");
    }
    else{
            printf("\n--------------------Employee'List--------------------");
    int i=0;
    while(i<=employeeSize){
            printf("\nName: %s\nID: %d\nNatinality: %s\nSex: %s\nAge: %d\nRating: %.2f\nMonthly Salary %.2f\n\n",emp[i].name,emp[i].id,emp[i].nationality,emp[i].gender,emp[i].age,emp[i].rating,emp[i].salary);

        i++;
    }
    }
```

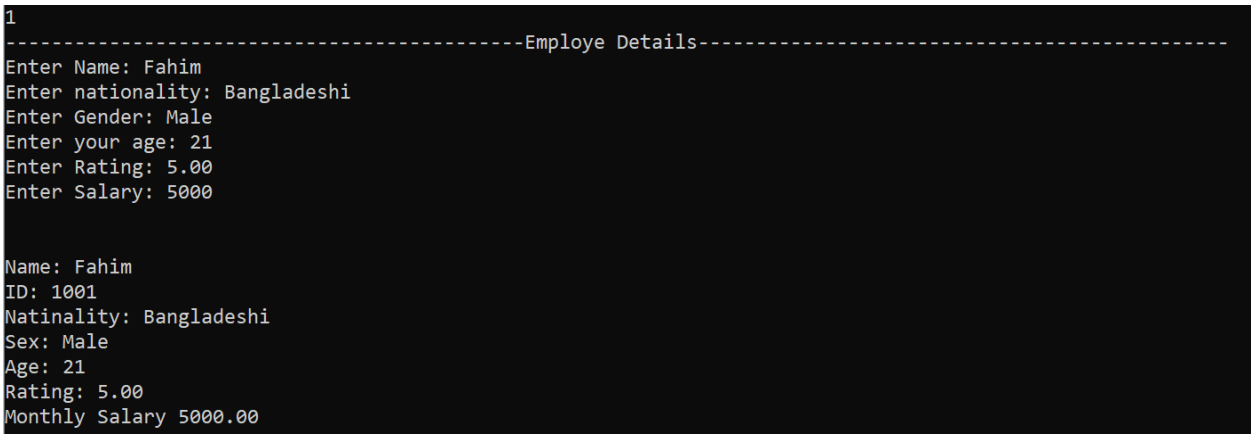
```


}
 void attendance()
{
    int attandanceCounter;
    char presentVar;

    printf("\n\nEmployee Attendance \n\n");

    for(attandanceCounter=0;attandanceCounter<=employeeSize;attandanceCounter++)
    {
        printf("(%d) Name: %s\tID: %d For Present press P for Absent press any ",attandanceCounter+1,emp[attandanceCounter].name,emp[attandanceCounter].id);
        scanf(" %c",&presentVar);


        if(tolower(presentVar)!='p'){
            continue;

        }

        emp[attandanceCounter].present+=1;

    }
}
 void employeedetails(employeeID)
{

    printf("\nName: %s\nID: %d\nNatin ality: %s\nSex: %s\nAge: %d\nRating: %.2f\nMonthly Salary %.2f\nPresent %d Days\n",emp[employeeID-1001].name,emp[employeeID-1001].id,emp[employeeID-1001].nationality,
        emp[employeeID-1001].gender,emp[employeeID-1001].age,emp[employeeID-1001].rating,emp[employeeID-1001].salary,emp[employeeID-1001].present);
    float taxAmount=saleryTaxcalculate(employeeID);

    printf("\nSalary without tax %.2f",emp[employeeID-1001].salary);
    printf("\nSalary with 6 Percent Gov Tax %.2f",(emp[employeeID-1001].salary) - taxAmount);

}
 int saleryTaxcalculate(employeeID)
{

    float tax=(emp[employeeID-1001].salary/100*6);



    return tax;
}
 void noticeFile()
{

    FILE *file;
    char ch;

    // Open the file in "r" (read) mode
    file = fopen("notice.txt", "r");
```

```


    if (file == NULL) {
        printf("Error opening the file.\n");
        return 1;
    }

    // Read and print each character until the end of the file is reached
    while ((ch = fgetc(file)) != EOF) {
        printf("%c", ch);
    }

    // Close the file
    fclose(file);

    getch();
}
```

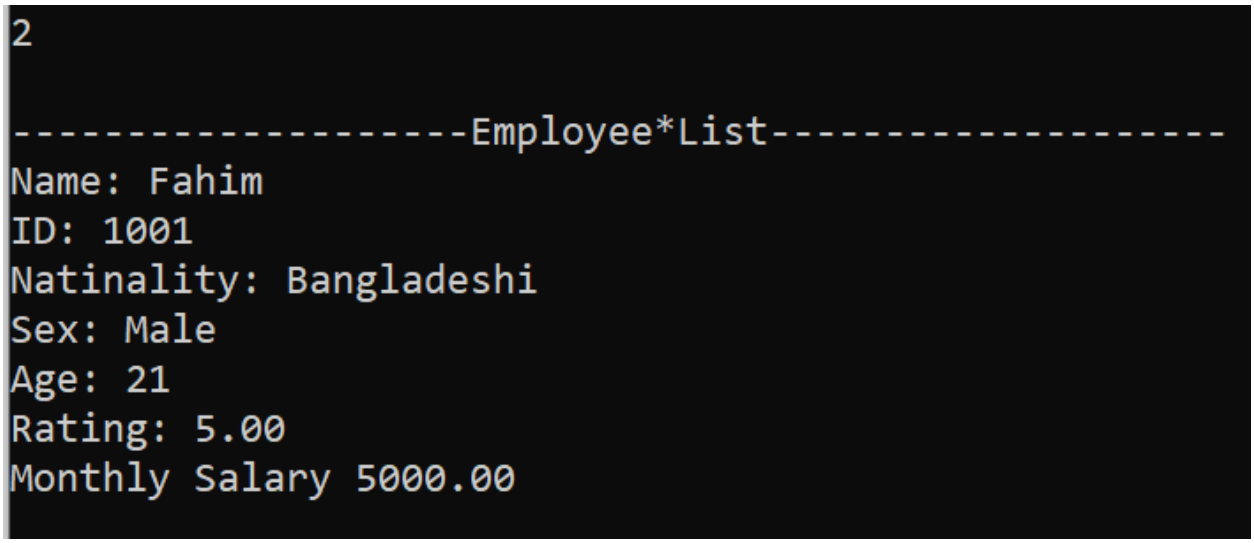
Sample Output



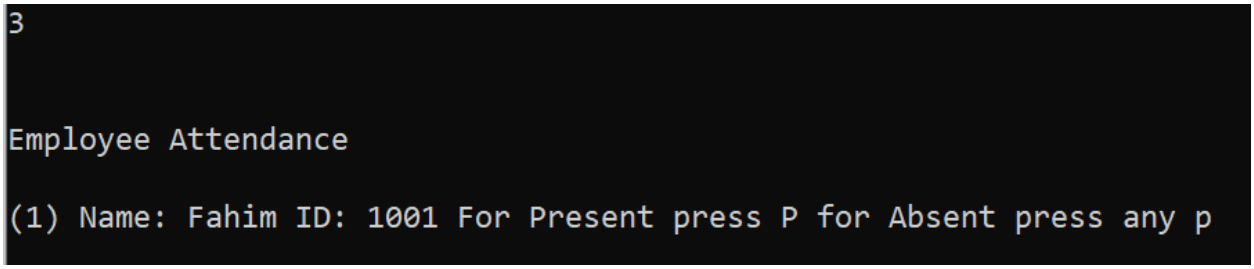
First the output will present the main menu for user choice.



1. After press 1 we can add employees.



2. After press 2 we can view the employee list how many employees we have.



3. after press 3 we can take the attendance for employees by press p

```
4
Enter Employee ID : 1001

Name: Fahim
ID: 1001
Natin ality: Bangladeshi
Sex: Male
Age: 21
Rating: 5.00
Monthly Salary 5000.00
Present 1 Days

Salary without tax 5000.00
Salary with 6 Percent Gov Tax 4700.00
```

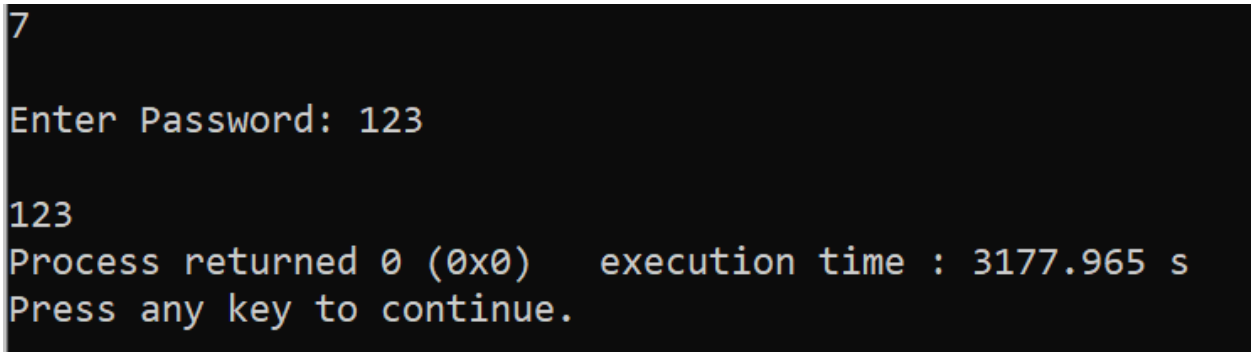
4. after press 4 we can find the employee full details through they are ID

```
5
Next Friday is Publice Holidays.
```

5. after press 5 it will show company notice

```
6

Reg.No: Fahim 1211205362
```

6. after press 6 it will show company registration number



7. After pressing 7 it will ask about password. If we input correct password our program will be end.

References

https://www.programiz.com/c-programming/c-file-input-output