NSU COURSE MANAGEMENT SYSTEM

INTRODUCTION :

This project introduces a prototype of a Student Management System, which we named, "NSU Course Management System." The whole project is completed using C programming language . In this project, a user can choose certain courses, which contains specific amount of credits. As per the user inputs, the program calculates course fees and provides it to the user. All the information provided by the user gets stored in a text file for further usages.

For example – if a user selects 4 courses, the program will calculate the total credits of the 4 courses and the fees per semester, and will display a final output of the courses user selected, total credits and the total fees.

Objectives

- Making course easier using interactive selection process .
- Implementing user friendly UI.
- Implementing the basic C programming knowledge we gathered throughout our course.
- Displaying sorted course list to make the selection procedure efficient.
- Practice analyzing and debugging techniques.
- Using separate user defined functions to make the code concise and modular.
- Writing comments where necessary to make the code understandable.
- Making cleaner alignment of the code to improve readability.
- Implementing efficient ways to handle errors , a user can face while running the program.

Team Configuration

We divided the program into 3 separate segments to complete the project. The segments are UI designing, main code structure, and error handling. All of the members of our group contributed to the main code structure and error handling. One of our group member handled the UI elements of the project.

OVERALL DESCRIPTION:

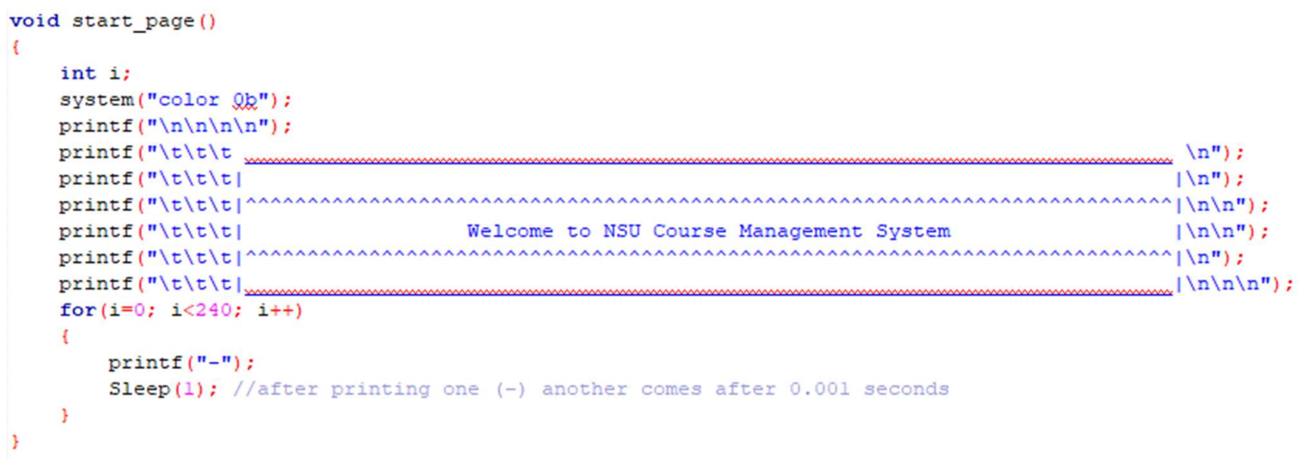
Code Mechanism :

In the beginning of the program, the user is greeted by a **welcome display** and encounters a **continuation message** asking if the user wants to continue further or not. As the user proceeds further, the program asks the user to input the user's **Name** and **ID**. After getting the name and ID input, these are stored in a **structure array** for further usages in the code. Then the code requests the user to type the **number of courses** the user desires to take. Depending on the number of courses, the user is prompted to the **course initial list**. From the list, the user has to input the corresponding number the initials are written into. According to the course initials, the program displays a list of that specific set of courses. The course lists are **pre-written** into a **text file**. When the user inputs a number, the program **reads** the corresponding course list file and displays it on the terminal. From the list, the user can choose any course by typing the course name in the respective field. The course user inputs every time, gets stored in a structure array and at the time the program **calculates** the **credits** of chosen courses. After the course selection procedure is done, the user gets an overview of the **chosen courses** , **total credits** and **the semester fees** . All of these information (Name, ID, Courses) gets stored in a text file named “**Student_info.txt**” using append mode of file handling.

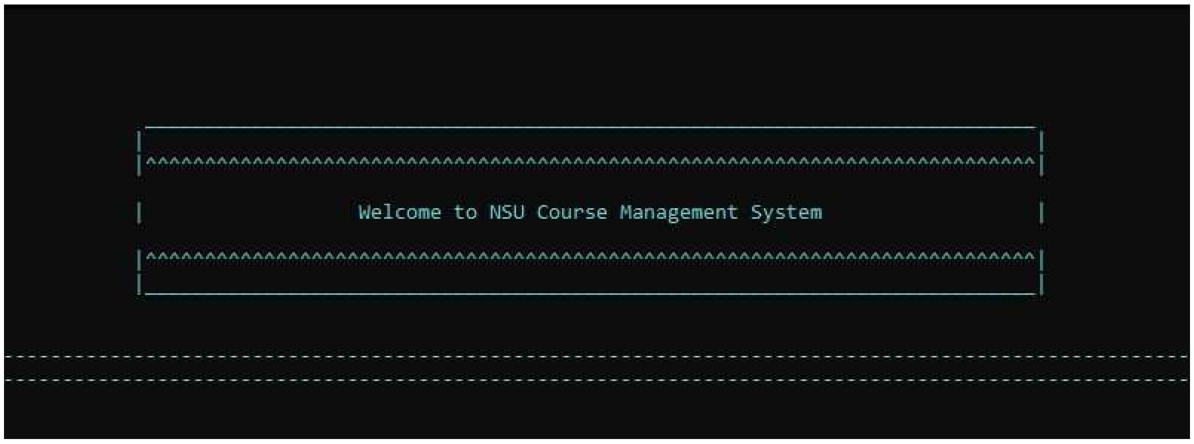
FUNCTIONS :

1. start_page() -

- Displays the welcome display and decorations of the program.
- By using the printf() function, UI elements are added.
- Using system("color 0b"), console font color is changed to aqua blue.



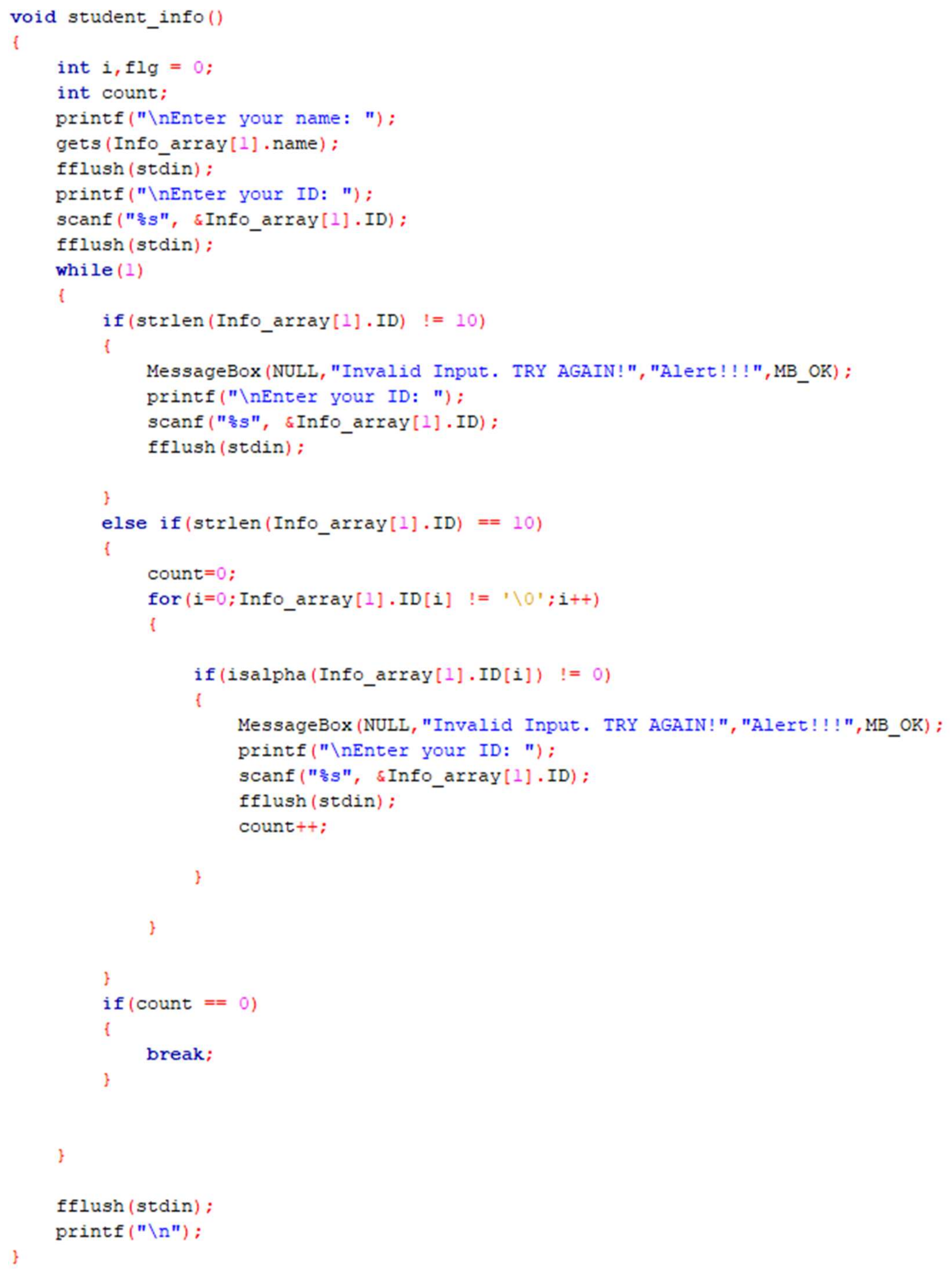
```c
void start_page()
{
    int i;
    system("color 0b");
    printf("\n\n\n\n");
    printf("\t\t\t ____________________________________________________________________________ \n");
    printf("\t\t\t|                                                                            |\n");
    printf("\t\t\t|^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^|\n\n");
    printf("\t\t\t|                 Welcome to NSU Course Management System                    |\n\n");
    printf("\t\t\t|^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^|\n");
    printf("\t\t\t|____________________________________________________________________________|\n\n\n");
    for(i=0; i<240; i++)
    {
        printf("-");
        Sleep(1); //after printing one (-) another comes after 0.001 seconds
    }
}
```



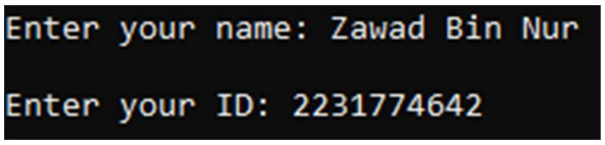
```
 ____________________________________________________________________________
|                                                                            |
|^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^|

|                 Welcome to NSU Course Management System                    |

|^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^|
|____________________________________________________________________________|


------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------------------
```

2. student_info() -

- Taking the input of student’s name and ID.
- In the structure array Info_array[ ], the name and ID are stored.



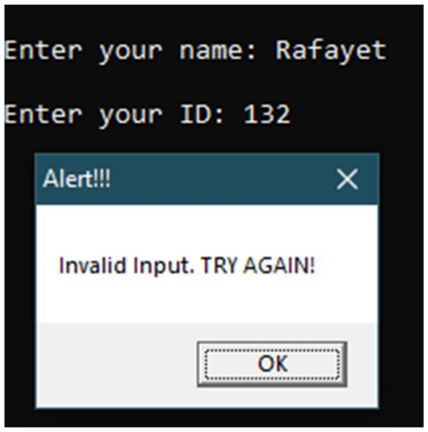
```c
void student_info()
{
    int i,flg = 0;
    int count;
    printf("\nEnter your name: ");
    gets(Info_array[1].name);
    fflush(stdin);
    printf("\nEnter your ID: ");
    scanf("%s", &Info_array[1].ID);
    fflush(stdin);
    while(1)
    {
        if(strlen(Info_array[1].ID) != 10)
        {
            MessageBox(NULL,"Invalid Input. TRY AGAIN!","Alert!!!",MB_OK);
            printf("\nEnter your ID: ");
            scanf("%s", &Info_array[1].ID);
            fflush(stdin);

        }
        else if(strlen(Info_array[1].ID) == 10)
        {
            count=0;
            for(i=0;Info_array[1].ID[i] != '\0';i++)
            {

                if(isalpha(Info_array[1].ID[i]) != 0)
                {
                    MessageBox(NULL,"Invalid Input. TRY AGAIN!","Alert!!!",MB_OK);
                    printf("\nEnter your ID: ");
                    scanf("%s", &Info_array[1].ID);
                    fflush(stdin);
                    count++;

                }

            }

        }
        if(count == 0)
        {
            break;
        }


    }

    fflush(stdin);
    printf("\n");
}
```



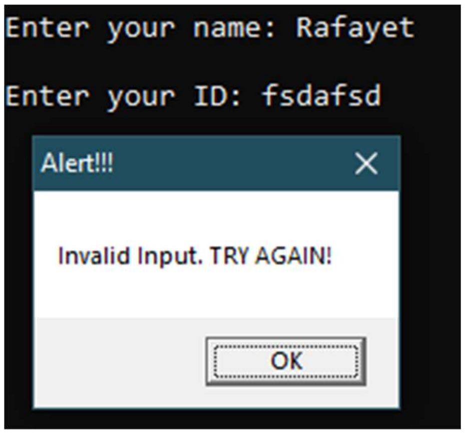
```
Enter your name: Zawad Bin Nur

Enter your ID: 2231774642
```

ERROR HANDLING –

- When the ID length is not equal to 10, program shows invalid input window.
- If user inputs a character type in ID, invalid input window pops up.
- In the loop, we are checking the string length of the ID using strlen().
- In the else if condition, we are checking if there are any characters.
- Every time the user provides a wrong input, a message box pops up and shows a message informing the user about it.



```
Enter your name: Rafayet

Enter your ID: 132
```

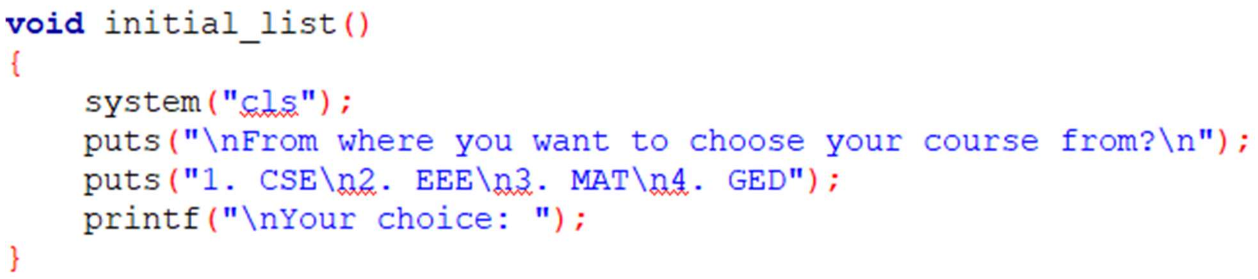


```
Enter your name: Rafayet

Enter your ID: fsdafsd
```

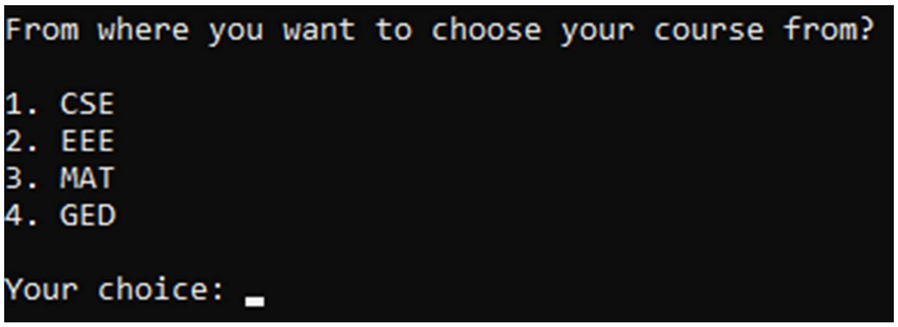
***P.S**: Whenever a user inputs an ID which is not 10 digits long or if the input contains a character in it, the program will show a message box informing the user about the invalid input.*

3. initial_list() -

- Using system("cls"), the console screen gets cleared.
- Using printf(), we are showing the course initial list.



```
void initial_list()
{
    system("cls");
    puts("\nFrom where you want to choose your course from?\n");
    puts("1. CSE\n2. EEE\n3. MAT\n4. GED");
    printf("\nYour choice: ");
}
```

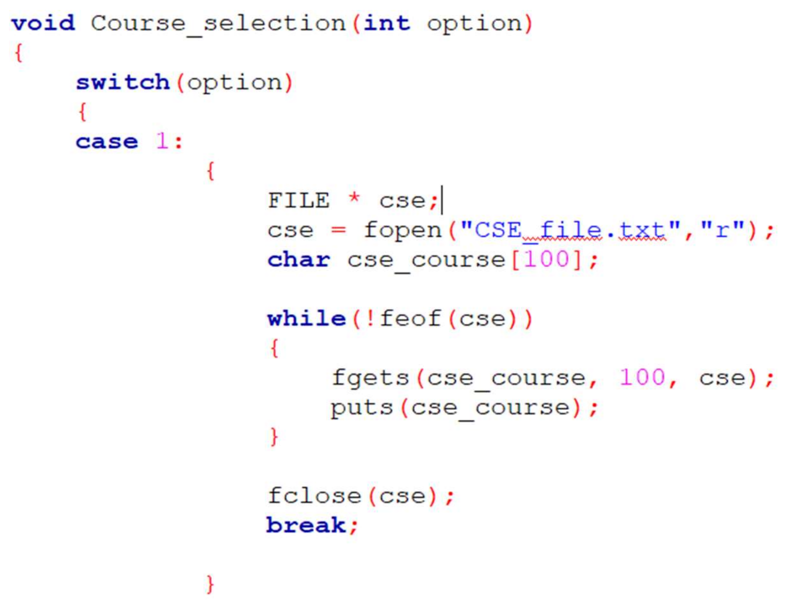


```
From where you want to choose your course from?

1. CSE
2. EEE
3. MAT
4. GED

Your choice: _
```

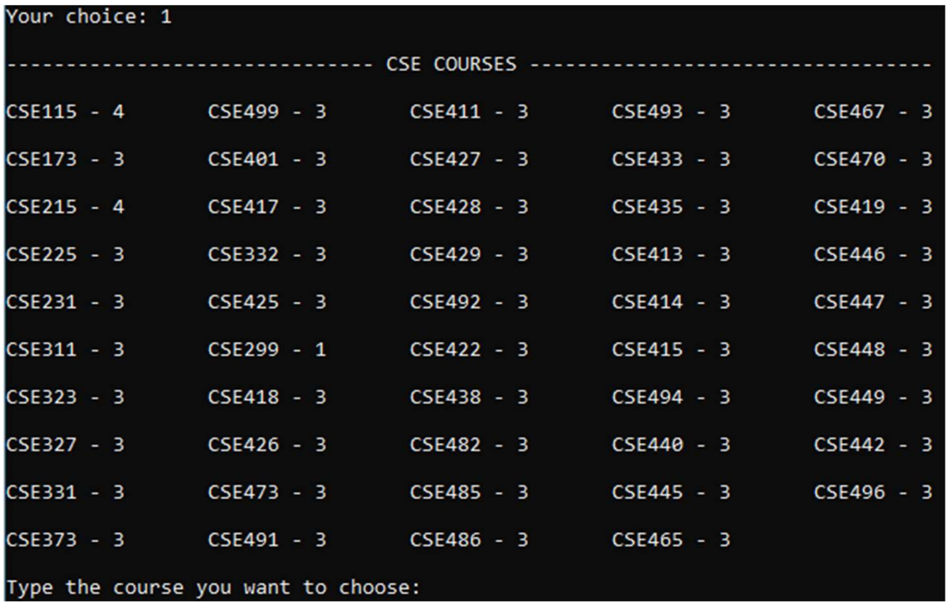
4. Course_selection(int option) -

- Taking input from the user, which determines which case to execute.
- Passing the argument into the function and opening respective text files.
- By using switch(option) and 4 cases, text file opening code is executed.

CASE-1: (Reads CSE_file.txt)

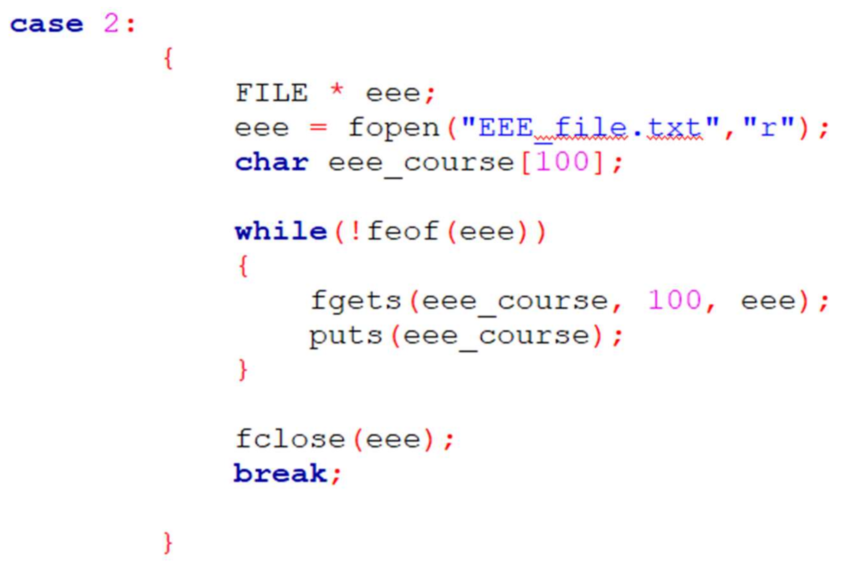


```
void Course_selection(int option)
{
    switch(option)
    {
    case 1:
            {
                FILE * cse;
                cse = fopen("CSE_file.txt","r");
                char cse_course[100];

                while(!feof(cse))
                {
                    fgets(cse_course, 100, cse);
                    puts(cse_course);
                }

                fclose(cse);
                break;

            }
```

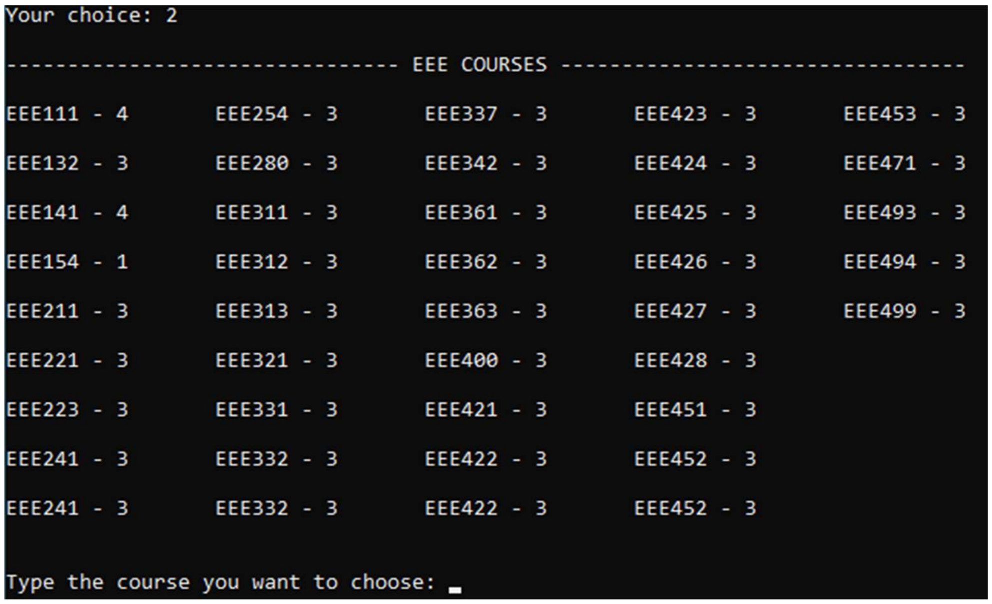


```
Your choice: 1

------------------------------ CSE COURSES ---------------------------------

CSE115 - 4      CSE499 - 3      CSE411 - 3      CSE493 - 3      CSE467 - 3

CSE173 - 3      CSE401 - 3      CSE427 - 3      CSE433 - 3      CSE470 - 3

CSE215 - 4      CSE417 - 3      CSE428 - 3      CSE435 - 3      CSE419 - 3

CSE225 - 3      CSE332 - 3      CSE429 - 3      CSE413 - 3      CSE446 - 3

CSE231 - 3      CSE425 - 3      CSE492 - 3      CSE414 - 3      CSE447 - 3

CSE311 - 3      CSE299 - 1      CSE422 - 3      CSE415 - 3      CSE448 - 3

CSE323 - 3      CSE418 - 3      CSE438 - 3      CSE494 - 3      CSE449 - 3

CSE327 - 3      CSE426 - 3      CSE482 - 3      CSE440 - 3      CSE442 - 3

CSE331 - 3      CSE473 - 3      CSE485 - 3      CSE445 - 3      CSE496 - 3

CSE373 - 3      CSE491 - 3      CSE486 - 3      CSE465 - 3

Type the course you want to choose:
```

CASE-2: (Reads EEE_file.txt)

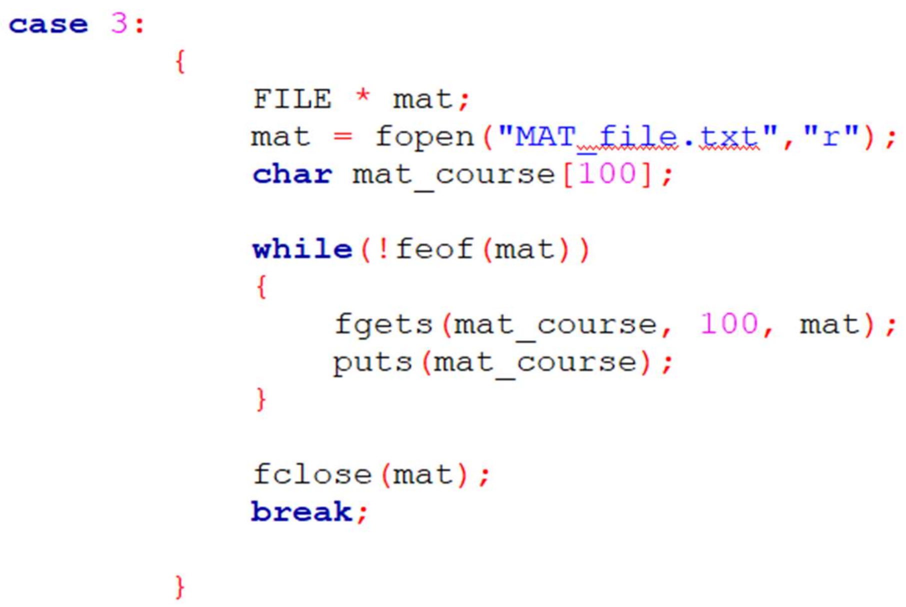


```
case 2:
        {
            FILE * eee;
            eee = fopen("EEE_file.txt","r");
            char eee_course[100];

            while(!feof(eee))
            {
                fgets(eee_course, 100, eee);
                puts(eee_course);
            }

            fclose(eee);
            break;

        }
```

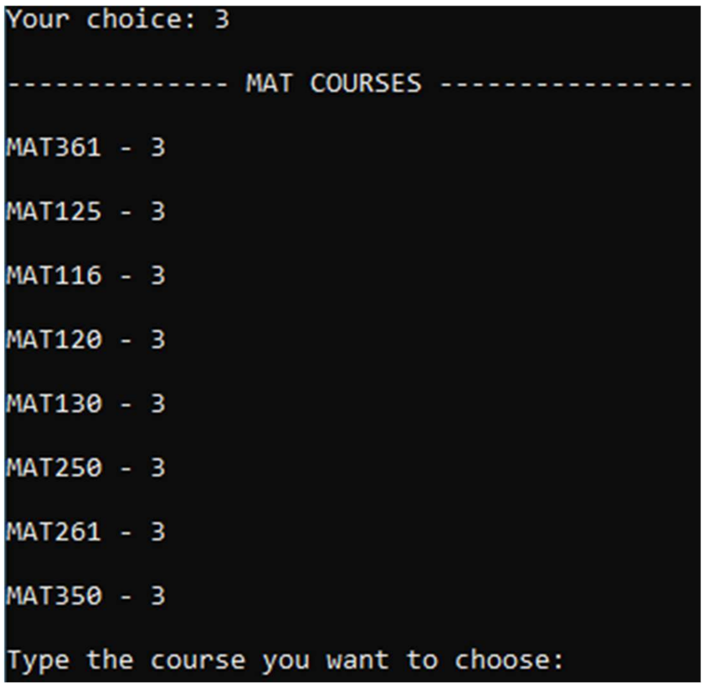


```
Your choice: 2

-------------------------------- EEE COURSES ---------------------------------

EEE111 - 4       EEE254 - 3       EEE337 - 3       EEE423 - 3       EEE453 - 3

EEE132 - 3       EEE280 - 3       EEE342 - 3       EEE424 - 3       EEE471 - 3

EEE141 - 4       EEE311 - 3       EEE361 - 3       EEE425 - 3       EEE493 - 3

EEE154 - 1       EEE312 - 3       EEE362 - 3       EEE426 - 3       EEE494 - 3

EEE211 - 3       EEE313 - 3       EEE363 - 3       EEE427 - 3       EEE499 - 3

EEE221 - 3       EEE321 - 3       EEE400 - 3       EEE428 - 3

EEE223 - 3       EEE331 - 3       EEE421 - 3       EEE451 - 3

EEE241 - 3       EEE332 - 3       EEE422 - 3       EEE452 - 3

EEE241 - 3       EEE332 - 3       EEE422 - 3       EEE452 - 3


Type the course you want to choose: _
```

CASE-3: (Reads MAT_file.txt)

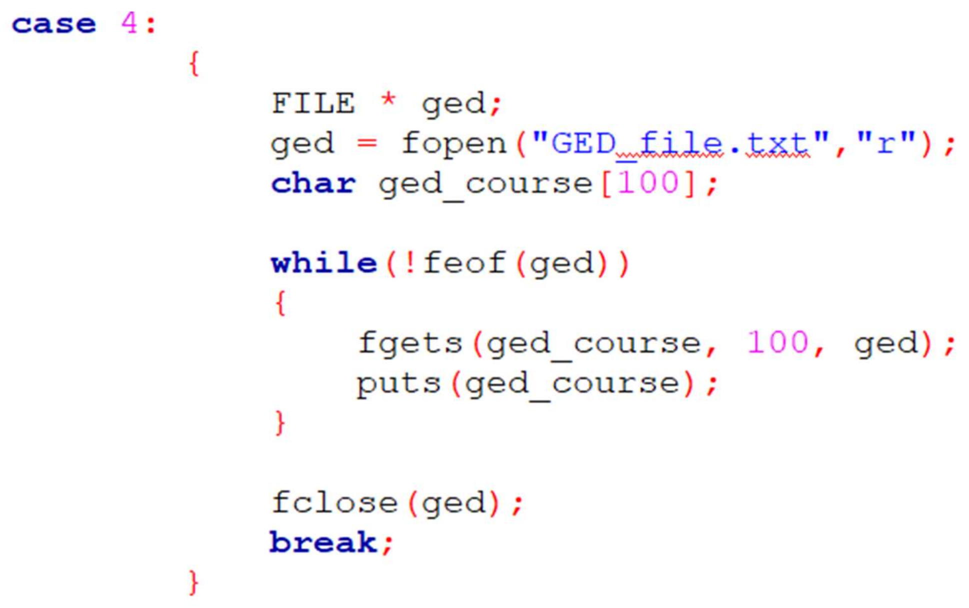


```
case 3:
        {
            FILE * mat;
            mat = fopen("MAT_file.txt","r");
            char mat_course[100];

            while(!feof(mat))
            {
                fgets(mat_course, 100, mat);
                puts(mat_course);
            }

            fclose(mat);
            break;

        }
```

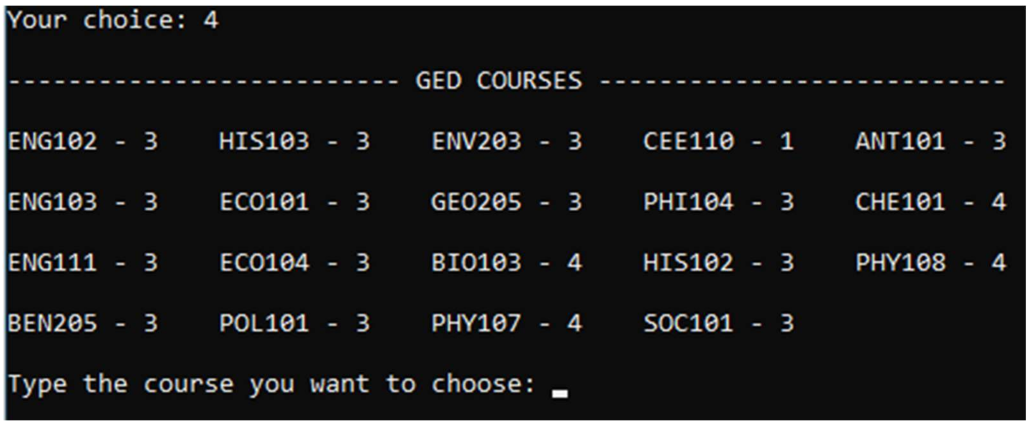


```
Your choice: 3

------------- MAT COURSES ---------------

MAT361 - 3

MAT125 - 3

MAT116 - 3

MAT120 - 3

MAT130 - 3

MAT250 - 3

MAT261 - 3

MAT350 - 3

Type the course you want to choose:
```

CASE-4: (Reads GED_file.txt)



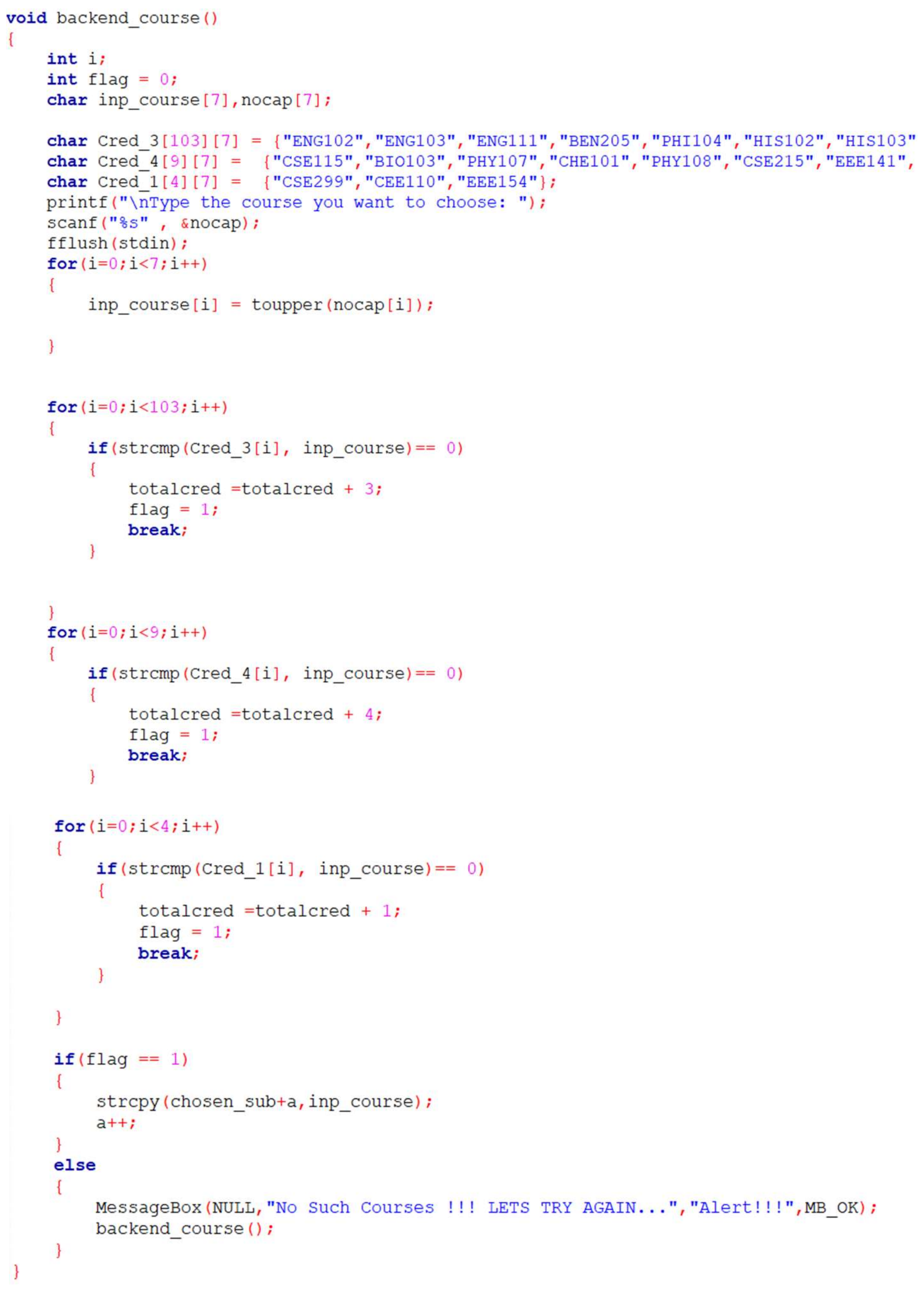
```c
case 4:
        {
            FILE * ged;
            ged = fopen("GED_file.txt","r");
            char ged_course[100];

            while(!feof(ged))
            {
                fgets(ged_course, 100, ged);
                puts(ged_course);
            }

            fclose(ged);
            break;
        }
```



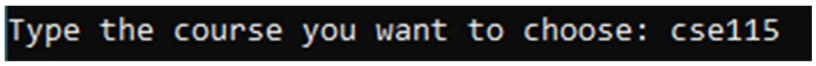
```
Your choice: 4

-------------------------- GED COURSES ---------------------------

ENG102 - 3    HIS103 - 3    ENV203 - 3    CEE110 - 1    ANT101 - 3

ENG103 - 3    ECO101 - 3    GEO205 - 3    PHI104 - 3    CHE101 - 4

ENG111 - 3    ECO104 - 3    BIO103 - 4    HIS102 - 3    PHY108 - 4

BEN205 - 3    POL101 - 3    PHY107 - 4    SOC101 - 3

Type the course you want to choose: _
```

5. backend_course() -

- Taking course name inputs.
- Storing course names into an array.
- In another array, we are storing the capitalized version of course names.
- By taking the course names, we are calculating total credits.
- We sorted the courses according to the credits into 3 different arrays.



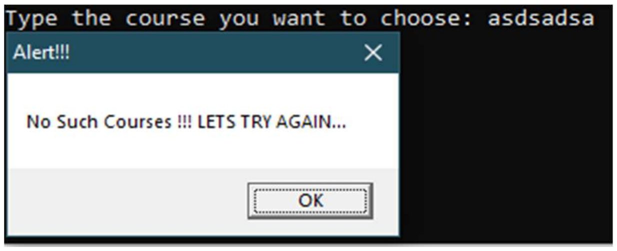
```
void backend_course()
{
    int i;
    int flag = 0;
    char inp_course[7],nocap[7];

    char Cred_3[103][7] = {"ENG102","ENG103","ENG111","BEN205","PHI104","HIS102","HIS103"
    char Cred_4[9][7] =  {"CSE115","BIO103","PHY107","CHE101","PHY108","CSE215","EEE141",
    char Cred_1[4][7] =  {"CSE299","CEE110","EEE154"};
    printf("\nType the course you want to choose: ");
    scanf("%s" , &nocap);
    fflush(stdin);
    for(i=0;i<7;i++)
    {
        inp_course[i] = toupper(nocap[i]);

    }


    for(i=0;i<103;i++)
    {
        if(strcmp(Cred_3[i], inp_course)== 0)
        {
            totalcred =totalcred + 3;
            flag = 1;
            break;
        }


    }
    for(i=0;i<9;i++)
    {
        if(strcmp(Cred_4[i], inp_course)== 0)
        {
            totalcred =totalcred + 4;
            flag = 1;
            break;
        }

     for(i=0;i<4;i++)
     {
         if(strcmp(Cred_1[i], inp_course)== 0)
         {
             totalcred =totalcred + 1;
             flag = 1;
             break;
         }

     }

     if(flag == 1)
     {
         strcpy(chosen_sub+a,inp_course);
         a++;
     }
     else
     {
         MessageBox(NULL,"No Such Courses !!! LETS TRY AGAIN...","Alert!!!",MB_OK);
         backend_course();
     }
 }
```



```
Type the course you want to choose: cse115
```

ERROR HANDLING –

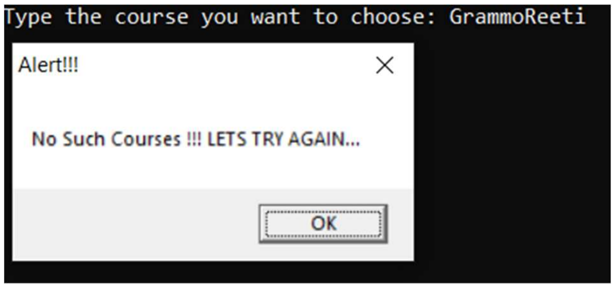
- After taking the course input, if the input and the course of declared array does not match, a message box pops up with a message - "No such Courses !!! LET'S TRY AGAIN…"



```
Type the course you want to choose: asdsadsa
```

Alert!!!

No Such Courses !!! LETS TRY AGAIN...



```
Type the course you want to choose: GrammoReeti
```

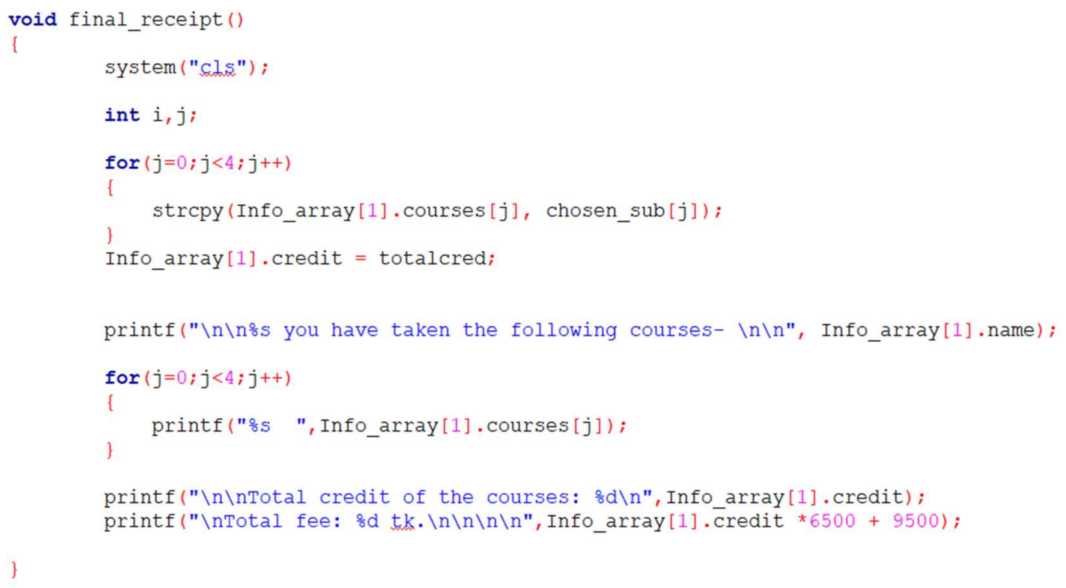
Alert!!!

No Such Courses !!! LETS TRY AGAIN...

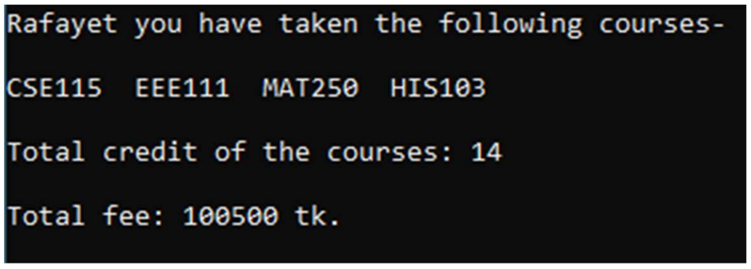
P.S: Whenever user inputs an invalid course, the user gets a pop-up message informing the user about the error.

6. final_receipt() –

- The courses user selected in the backend_course() function, gets stored in the structure array.
- The total credit we get from backend_course() function also gets stored in the structure array.
- Then finally it prints a receipt of all the courses the user chose, total credits and the semester fees (6500 tk. per credit + 9500 tk. Activity fee)



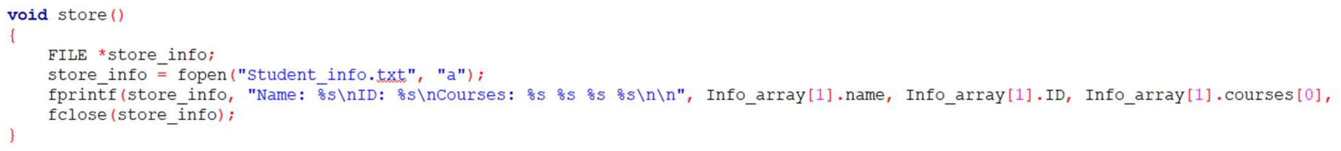
```c
void final_receipt()
{
        system("cls");

        int i,j;

        for(j=0;j<4;j++)
        {
            strcpy(Info_array[1].courses[j], chosen_sub[j]);
        }
        Info_array[1].credit = totalcred;


        printf("\n\n%s you have taken the following courses- \n\n", Info_array[1].name);

        for(j=0;j<4;j++)
        {
            printf("%s  ",Info_array[1].courses[j]);
        }

        printf("\n\nTotal credit of the courses: %d\n",Info_array[1].credit);
        printf("\nTotal fee: %d tk.\n\n\n\n",Info_array[1].credit *6500 + 9500);

}
```



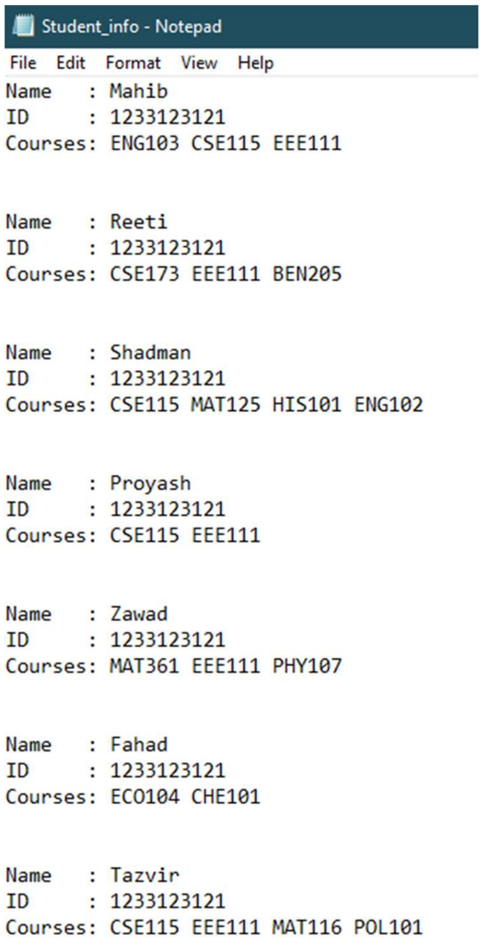
```
Rafayet you have taken the following courses-

CSE115  EEE111  MAT250  HIS103

Total credit of the courses: 14

Total fee: 100500 tk.
```

7. store() -

- Declaring file pointer.
- Opening a file named "Student_info.txt" and using the append mode of file handling, storing the Names, IDs and Courses.



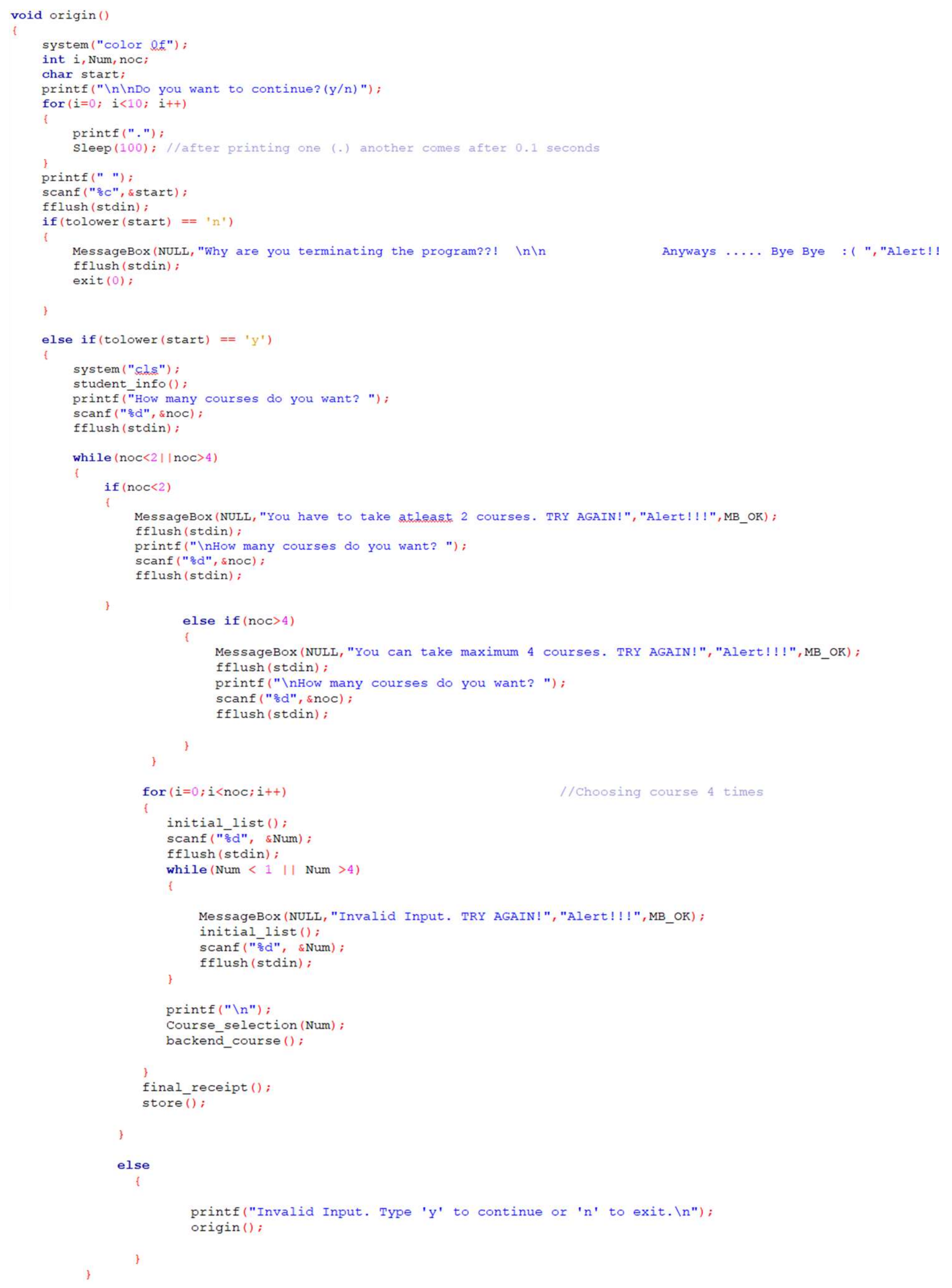
```c
void store()
{
    FILE *store_info;
    store_info = fopen("Student_info.txt", "a");
    fprintf(store_info, "Name: %s\nID: %s\nCourses: %s %s %s %s\n\n", Info_array[1].name, Info_array[1].ID, Info_array[1].courses[0],
    fclose(store_info);
}
```



```
Name    : Mahib
ID      : 1233123121
Courses: ENG103 CSE115 EEE111


Name    : Reeti
ID      : 1233123121
Courses: CSE173 EEE111 BEN205


Name    : Shadman
ID      : 1233123121
Courses: CSE115 MAT125 HIS101 ENG102


Name    : Proyash
ID      : 1233123121
Courses: CSE115 EEE111


Name    : Zawad
ID      : 1233123121
Courses: MAT361 EEE111 PHY107


Name    : Fahad
ID      : 1233123121
Courses: ECO104 CHE101


Name    : Tazvir
ID      : 1233123121
Courses: CSE115 EEE111 MAT116 POL101
```

8. origin() –

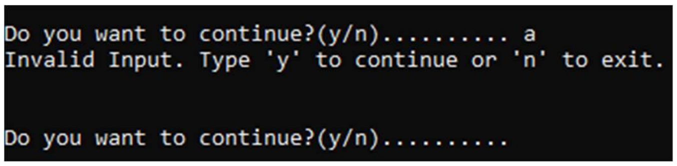
- Showing a program continuation message.
- If the program moves to further stage, it calls student_info() function.
- Takes the input of the number of courses user wants.
- Initial_list() function is called by origin().
- Taking input from user, the course_selection() function is called.
- After taking all the inputs in course_selection() function, origin() calls the backend_course() function.
- Finally it calls final_receipt() and store() functions.



```c
void origin()
{
    system("color 0f");
    int i,Num,noc;
    char start;
    printf("\n\nDo you want to continue?(y/n)");
    for(i=0; i<10; i++)
    {
        printf(".");
        Sleep(100); //after printing one (.) another comes after 0.1 seconds
    }
    printf(" ");
    scanf("%c",&start);
    fflush(stdin);
    if(tolower(start) == 'n')
    {
        MessageBox(NULL,"Why are you terminating the program??!  \n\n              Anyways ..... Bye Bye  :( ","Alert!!
        fflush(stdin);
        exit(0);

    }

    else if(tolower(start) == 'y')
    {
        system("cls");
        student_info();
        printf("How many courses do you want? ");
        scanf("%d",&noc);
        fflush(stdin);

        while(noc<2||noc>4)
        {
            if(noc<2)
            {
                MessageBox(NULL,"You have to take atleast 2 courses. TRY AGAIN!","Alert!!!",MB_OK);
                fflush(stdin);
                printf("\nHow many courses do you want? ");
                scanf("%d",&noc);
                fflush(stdin);

            }
                    else if(noc>4)
                    {
                        MessageBox(NULL,"You can take maximum 4 courses. TRY AGAIN!","Alert!!!",MB_OK);
                        fflush(stdin);
                        printf("\nHow many courses do you want? ");
                        scanf("%d",&noc);
                        fflush(stdin);

                    }
              }

              for(i=0;i<noc;i++)                              //Choosing course 4 times
              {
                  initial_list();
                  scanf("%d", &Num);
                  fflush(stdin);
                  while(Num < 1 || Num >4)
                  {

                      MessageBox(NULL,"Invalid Input. TRY AGAIN!","Alert!!!",MB_OK);
                      initial_list();
                      scanf("%d", &Num);
                      fflush(stdin);
                  }

                  printf("\n");
                  Course_selection(Num);
                  backend_course();

              }
              final_receipt();
              store();

          }

          else
            {

                  printf("Invalid Input. Type 'y' to continue or 'n' to exit.\n");
                  origin();

            }
      }
```

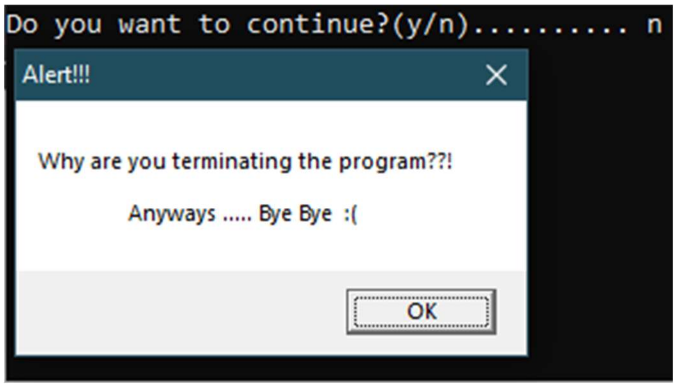
ERROR HANDLING –

- When user provides an input except 'y' or 'n', the program shows a message box of invalid input.
- While choosing number of courses, if the user inputs a digit (less than 2 or greater than 4) or character, a message box of invalid input pops up.
- After course_selection() function is called, if user provides an input outside (1-4), the programs shows invalid input message box.



```
Do you want to continue?(y/n).......... a
Invalid Input. Type 'y' to continue or 'n' to exit.

Do you want to continue?(y/n)..........
```

***P.S**: When user inputs any other character except 'y' or 'n' , the program prompts the user to retry.*



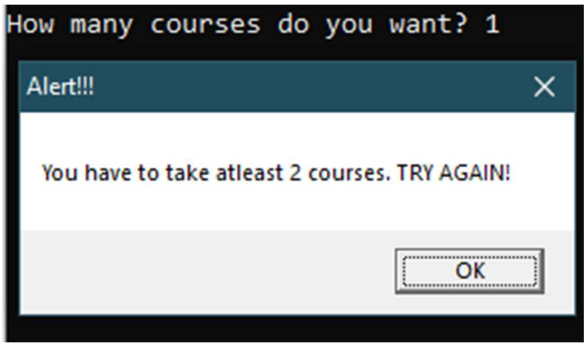
```
Do you want to continue?(y/n).......... n
```

Alert!!!

Why are you terminating the program??!

Anyways ..... Bye Bye :(

***P.S**: In case the user does not want to continue into the program, the program displays a farewell message.*

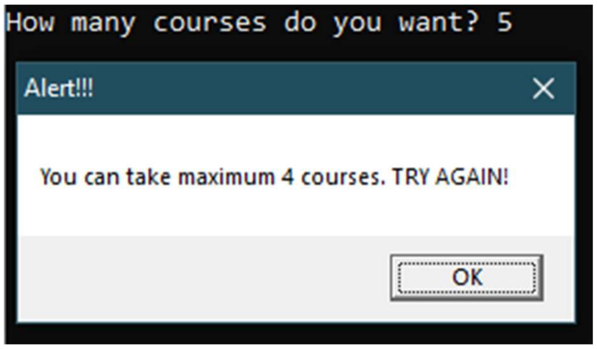


```
How many courses do you want? 1
```

Alert!!!

You have to take atleast 2 courses. TRY AGAIN!

P.S: *If the user chooses less than 2 courses the program tells the user that the user needs to choose at least 2 courses.*

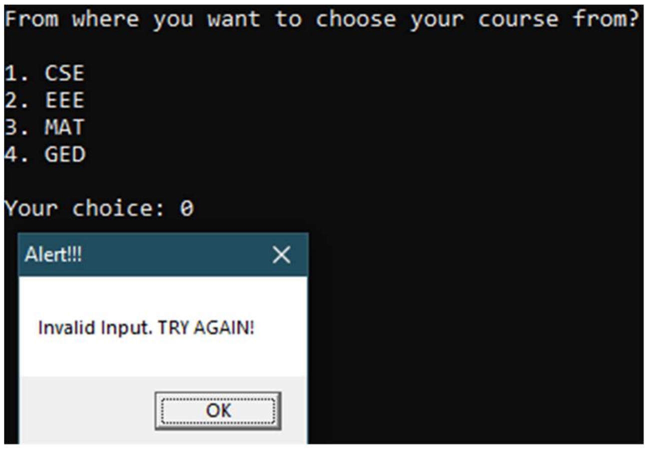


```
How many courses do you want? 5
```

Alert!!!

You can take maximum 4 courses. TRY AGAIN!

P.S: *If the user chooses more than 4 courses the program tells the user that the user can choose maximum 4 courses.*



```
From where you want to choose your course from?

1. CSE
2. EEE
3. MAT
4. GED

Your choice: 0
```

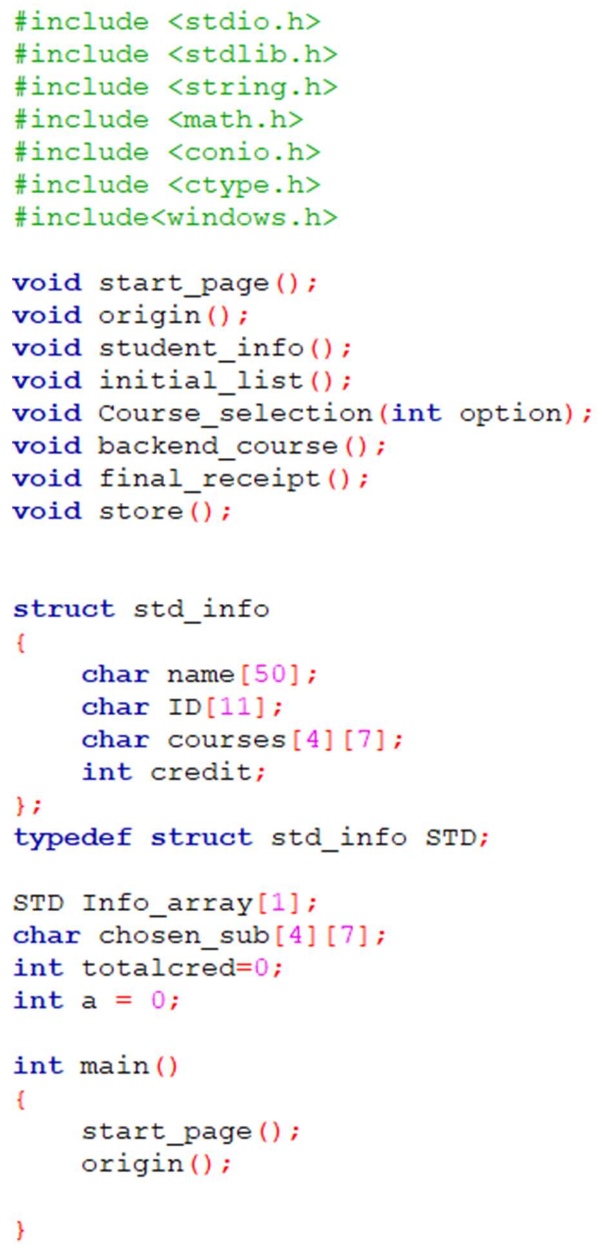
Alert!!!

Invalid Input. TRY AGAIN!

P.S: *If the input is anything outside the declared case numbers, the user will get an invalid input message.*

9. main() –

- The main function mainly calls the start_page() and origin() functions.



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <conio.h>
#include <ctype.h>
#include<windows.h>

void start_page();
void origin();
void student_info();
void initial_list();
void Course_selection(int option);
void backend_course();
void final_receipt();
void store();


struct std_info
{
    char name[50];
    char ID[11];
    char courses[4][7];
    int credit;
};
typedef struct std_info STD;

STD Info_array[1];
char chosen_sub[4][7];
int totalcred=0;
int a = 0;

int main()
{
    start_page();
    origin();

}
```

CONCLUSION :

The project is basically designed to help the students with the course selection conveniently. In addition to that, it computes the credits and semester fees providing an idea of the expenditures. Moreover, the program stores the student information for further usage.

Future scopes >>>

The project is prototype of the whole student management system. It can be implemented in any student management system. Plenty of features can be added to this project such as –

- Credentials management of the students.
- Faculty & Section selection.
- Database management.
- Transaction authentication.
- Grade Calculating System.
- Library check ins.
- Automated message delivery system.
- Course Mapping.
- Leaderboard integration.
- Fetching specific statistical data.
- Student counselling appointment.
- Direct messaging to faculties protocol.