ASSIGNMENT

By

SAMBHAV TICKOO

2023A7R014

2ND SEMSTER

CSE (CYBER SECURITY)

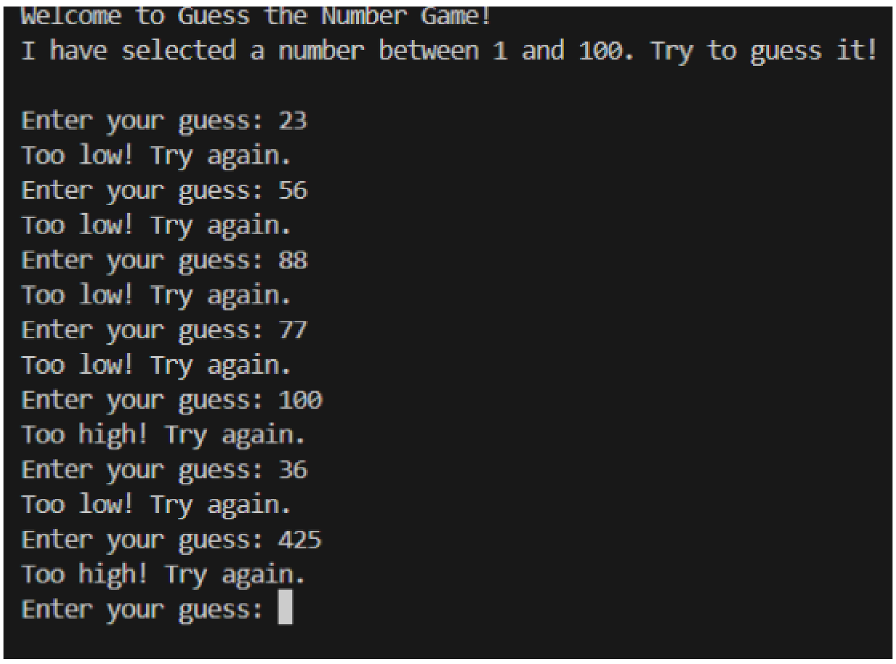
Model Institute of Engineering & Technology (Autonomous)

(Permanently Affiliated to the University of Jammu, Accredited by NAAC with “A” Grade) Jammu, India

2023

1.Develop a "Guess the Number" game with specified features: random number generation, user input validation, feedback on guesses, attempt limitation, game termination conditions, and score tracking

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>


// Function to generate a random number between a given range

int generateRandomNumber(int min, int max) {

    return rand() % (max - min + 1) + min;

}

// Function to play the guess the number game

void playGuessTheNumberGame() {

    const int minNumber = 1;

    const int maxNumber = 100;

    const int targetNumber = generateRandomNumber(minNumber, maxNumber);

    int guess;

    int attempts = 0;

    printf("Welcome to Guess the Number Game!\n");

    printf("I have selected a number between %d and %d. Try to guess it!\n\n",
minNumber, maxNumber);

    do {
        // Get user input

        printf("Enter your guess: ");

        scanf("%d", &guess);

        // Check if the guess is correct, too low, or too high

        if (guess == targetNumber) {

            printf("Congratulations! You guessed the number in %d attempts.\n", attempts
+ 1);

            break;

        } else if (guess < targetNumber) {

            printf("Too low! Try again.\n");

        } else {

            printf("Too high! Try again.\n");

        }

        attempts++;

    } while (1);

}

int main() {

    // Seed the random number generator with the current time

    srand(time(NULL));

    // Call the function to play the game
    playGuessTheNumberGame();

    return 0;

}
```



```
Welcome to Guess the Number Game!
I have selected a number between 1 and 100. Try to guess it!

Enter your guess: 23
Too low! Try again.
Enter your guess: 56
Too low! Try again.
Enter your guess: 88
Too low! Try again.
Enter your guess: 77
Too low! Try again.
Enter your guess: 100
Too high! Try again.
Enter your guess: 36
Too low! Try again.
Enter your guess: 425
Too high! Try again.
Enter your guess: 
```

2. Create a Library Management System with features for book and patron management, as well as borrowing and return processes, to streamline library operations effectively.

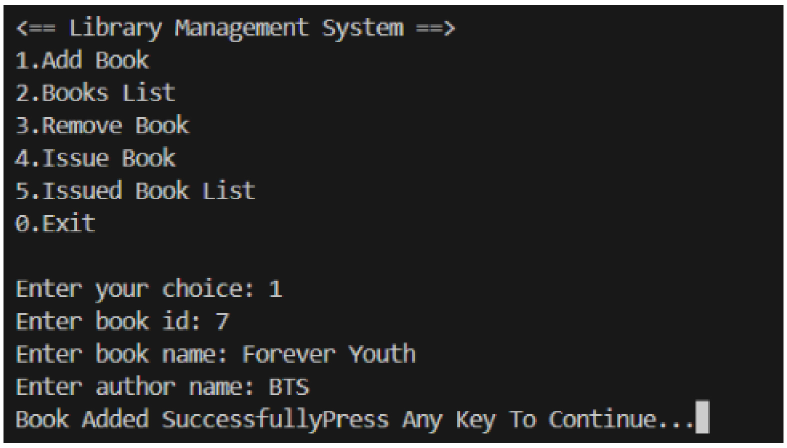
```
#include<stdio.h>


#include<stdlib.h>


#include<time.h>


struct books{


    int id;


    char bookName[50];
    char authorName[50];

    char date[12];

}b;

struct student{

    int id;

    char sName[50];

    char sClass[50];

    int sRoll;

    char bookName[50];

    char date[12];

}s;

FILE *fp;

int main(){

    int ch;

    while(1){

        system("cls");

        printf("<== Library Management System ==>\n");

        printf("1.Add Book\n");

        printf("2.Books List\n");
        printf("3.Remove Book\n");

        printf("4.Issue Book\n");

        printf("5.Issued Book List\n");

        printf("0.Exit\n\n");

        printf("Enter your choice: ");

        scanf("%d", &ch);

        switch(ch){

        case 0:

           exit(0);

        case 1:

           addBook();

           break;

        case 2:

           booksList();

           break;

        case 3:

           del();

           break;

        case 4:
```

```
            issueBook();

            break;

        case 5:

            issueList();

            break;

        default:

            printf("Invalid Choice...\n\n");

        }

        printf("Press Any Key To Continue...");

        getch();

    }

    return 0;

}

void addBook(){

    char myDate[12];

    time_t t = time(NULL);

    struct tm tm = *localtime(&t);

    sprintf(myDate, "%02d/%02d/%d", tm.tm_mday, tm.tm_mon+1, tm.tm_year +
1900);
```

```
    strcpy(b.date, myDate);

    fp = fopen("books.txt", "ab");

    printf("Enter book id: ");

    scanf("%d", &b.id);

    printf("Enter book name: ");

    fflush(stdin);

    gets(b.bookName);

    printf("Enter author name: ");

    fflush(stdin);

    gets(b.authorName);

    printf("Book Added Successfully");

    fwrite(&b, sizeof(b), 1, fp);

    fclose(fp);

}

void booksList(){

    system("cls");

    printf("<== Available Books ==>\n\n");

    printf("%-10s %-30s %-20s %s\n\n", "Book id", "Book Name", "Author", "Date");

    fp = fopen("books.txt", "rb");
    while(fread(&b, sizeof(b), 1, fp) == 1){

        printf("%-10d %-30s %-20s %s\n", b.id, b.bookName, b.authorName, b.date);

    }

    fclose(fp);

}

void del(){

    int id, f=0;

    system("cls");

    printf("<== Remove Books ==>\n\n");

    printf("Enter Book id to remove: ");

    scanf("%d", &id);

    FILE *ft;

    fp = fopen("books.txt", "rb");

    ft = fopen("temp.txt", "wb");

    while(fread(&b, sizeof(b), 1, fp) == 1){

        if(id == b.id){

            f=1;

        }else{

            fwrite(&b, sizeof(b), 1, ft);
        }

    }

    if(f==1){

        printf("\n\nDeleted Successfully.");

    }else{

        printf("\n\nRecord Not Found !");

    }

    fclose(fp);

    fclose(ft);

    remove("books.txt");

    rename("temp.txt", "books.txt");

}

void issueBook(){

    char myDate[12];

    time_t t = time(NULL);

    struct tm tm = *localtime(&t);

    sprintf(myDate, "%02d/%02d/%d", tm.tm_mday, tm.tm_mon+1, tm.tm_year +
1900);

    strcpy(s.date, myDate);
```

```
int f=0;

system("cls");

printf("<== Issue Books ==>\n\n");

printf("Enter Book id to issue: ");

scanf("%d", &s.id);

//Check if we have book of given id

fp = fopen("books.txt", "rb");

while(fread(&b, sizeof(b), 1, fp) == 1){

   if(b.id == s.id){

      strcpy(s.bookName, b.bookName);

      f=1;

      break;

   }

}

if(f==0){

   printf("No book found with this id\n");

   printf("Please try again...\n\n");

   return;

}
```

```
    fp = fopen("issue.txt", "ab");

    printf("Enter Student Name: ");

    fflush(stdin);

    gets(s.sName);

    printf("Enter Student Class: ");

    fflush(stdin);

    gets(s.sClass);

    printf("Enter Student Roll: ");

    scanf("%d", &s.sRoll);

    printf("Book Issued Successfully\n\n");

    fwrite(&s, sizeof(s), 1, fp);

    fclose(fp);

}

void issueList(){

    system("cls");

    printf("<== Book Issue List ==>\n\n");

    printf("%-10s %-30s %-20s %-10s %-30s %s\n\n", "S.id", "Name", "Class", "Roll",
"Book Name", "Date");

    fp = fopen("issue.txt", "rb");
    while(fread(&s, sizeof(s), 1, fp) == 1){

        printf("%-10d %-30s %-20s %-10d %-30s %s\n", s.id, s.sName, s.sClass, s.sRoll,
s.bookName, s.date);

    }

    fclose(fp);

}
```



```
<== Library Management System ==>
1.Add Book
2.Books List
3.Remove Book
4.Issue Book
5.Issued Book List
0.Exit

Enter your choice: 1
Enter book id: 7
Enter book name: Forever Youth
Enter author name: BTS
Book Added SuccessfullyPress Any Key To Continue...
```

3. Create a Student Grade Tracker, each with specified features to streamline library operations or manage student grades effectively.

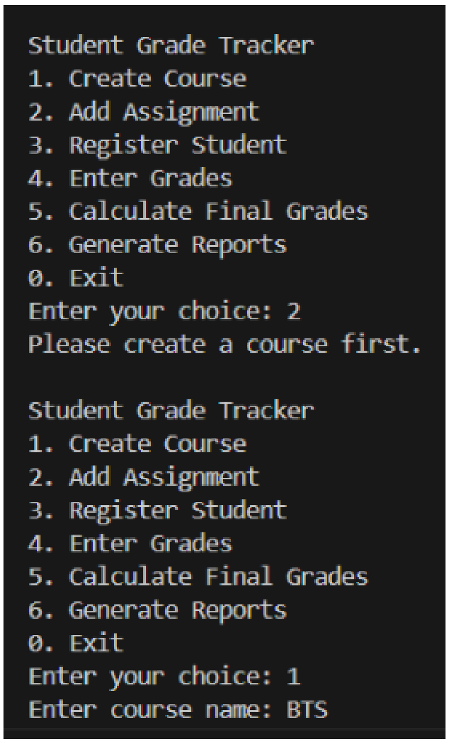
```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX_COURSES 10

#define MAX_ASSIGNMENTS 20

#define MAX_STUDENTS 50

// Structure definitions
```

```
typedef struct {

    char name[100];

    int assignmentCount;

    float assignmentScores[MAX_STUDENTS][MAX_ASSIGNMENTS];

    float weights[MAX_ASSIGNMENTS];

} Course;

typedef struct {

    int id;

    char name[100];

    float finalGrade;

} Student;

// Function prototypes

void createCourse(Course courses[], int *numCourses);

void addAssignment(Course *course);

void registerStudent(Student students[], int *numStudents);

void enterGrades(Course *course, Student students[], int numStudents);

void calculateFinalGrades(Course *course, int numStudents);

void generateReports(Course courses[], int numCourses, Student students[], int
numStudents);
```

```
int main() {
    Course courses[MAX_COURSES];
    Student students[MAX_STUDENTS];
    int numCourses = 0, numStudents = 0;
    int choice;
    do {
        printf("\nStudent Grade Tracker\n");
        printf("1. Create Course\n");
        printf("2. Add Assignment\n");
        printf("3. Register Student\n");
        printf("4. Enter Grades\n");
        printf("5. Calculate Final Grades\n");
        printf("6. Generate Reports\n");
        printf("0. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch(choice) {
            case 1:
                createCourse(courses, &numCourses);
                break;
            case 2:
                if (numCourses > 0) {
                    addAssignment(&courses[numCourses - 1]);
                } else {
                    printf("Please create a course first.\n");
                }
                break;
            case 3:
                registerStudent(students, &numStudents);
                break;
            case 4:
                if (numCourses > 0 && numStudents > 0) {
                    enterGrades(&courses[numCourses - 1], students, numStudents);
                } else {
                    printf("Please create a course and register students first.\n");
                }
                break;
            case 5:
                if (numCourses > 0 && numStudents > 0) {

                    calculateFinalGrades(&courses[numCourses - 1], numStudents);

                } else {

                    printf("Please create a course and register students first.\n");

                }

                break;

            case 6:

                if (numCourses > 0 && numStudents > 0) {

                    generateReports(courses, numCourses, students, numStudents);

                } else {

                    printf("Please create a course and register students first.\n");

                }

                break;

            case 0:

                printf("Exiting...\n");

                break;

            default:

                printf("Invalid choice. Please try again.\n");

        }
```

```
    } while (choice != 0);

    return 0;

}

void createCourse(Course courses[], int *numCourses) {

    if (*numCourses < MAX_COURSES) {

        printf("Enter course name: ");

        scanf("%s", courses[*numCourses].name);

        printf("Enter number of assignments: ");

        scanf("%d", &courses[*numCourses].assignmentCount);

        for (int i = 0; i < courses[*numCourses].assignmentCount; i++) {

            printf("Enter weight for assignment %d: ", i + 1);

            scanf("%f", &courses[*numCourses].weights[i]);

        }

        (*numCourses)++;

        printf("Course created successfully.\n");

    } else {

        printf("Maximum courses reached.\n");

    }

}
```

```
void addAssignment(Course *course) {

   if (course->assignmentCount < MAX_ASSIGNMENTS) {

      printf("Enter assignment name: ");

      scanf("%s", course->name);

      (course->assignmentCount)++;

      printf("Assignment added successfully.\n");

   } else {

      printf("Maximum assignments reached for this course.\n");

   }

}

void registerStudent(Student students[], int *numStudents) {

   if (*numStudents < MAX_STUDENTS) {

      printf("Enter student ID: ");

      scanf("%d", &students[*numStudents].id);

      printf("Enter student name: ");

      scanf("%s", students[*numStudents].name);

      (*numStudents)++;

      printf("Student registered successfully.\n");

   } else {
        printf("Maximum students reached.\n");
    }
}
void enterGrades(Course *course, Student students[], int numStudents) {
    for (int i = 0; i < numStudents; i++) {
        printf("Enter grades for student %s:\n", students[i].name);
        for (int j = 0; j < course->assignmentCount; j++) {
            printf("Enter grade for assignment %d: ", j + 1);
            scanf("%f", &course->assignmentScores[i][j]);
        }
    }
    printf("Grades entered successfully.\n");
}
void calculateFinalGrades(Course *course, int numStudents) {
    for (int i = 0; i < numStudents; i++) {
        float finalGrade = 0;
        for (int j = 0; j < course->assignmentCount; j++) {
            finalGrade += course->assignmentScores[i][j] * course->weights[j];
        }
        course->assignmentScores[i][course->assignmentCount] = finalGrade;

    }

    printf("Final grades calculated successfully.\n");

}

void generateReports(Course courses[], int numCourses, Student students[], int
numStudents) {

    // Report on individual student performance

    printf("Student Performance Report:\n");

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

        printf("Student ID: %d, Name: %s, Final Grade: %.2f\n", students[i].id,
students[i].name, students[i].finalGrade);

    }

    printf("\n");

    // Report on class averages

    printf("Class Averages Report:\n");

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

        float totalGrade = 0;

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

            totalGrade += courses[i].weights[j];

        }
        float classAverage = totalGrade / numStudents;

        printf("Course: %s, Class Average: %.2f\n", courses[i].name, classAverage);

    }

    printf("\n");

    // Report on grade distribution

    printf("Grade Distribution Report:\n");

    int gradeCounts[5] = {0}; // Assuming grading scale of 5 levels

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

        if (students[i].finalGrade >= 90) {

            gradeCounts[0]++;

        } else if (students[i].finalGrade >= 80) {

            gradeCounts[1]++;

        } else if (students[i].finalGrade >= 70) {

            gradeCounts[2]++;

        } else if (students[i].finalGrade >= 60) {

            gradeCounts[3]++;

        } else {

            gradeCounts[4]++;

        }
    }

    printf("A (90-100): %d\n", gradeCounts[0]);

    printf("B (80-89): %d\n", gradeCounts[1]);

    printf("C (70-79): %d\n", gradeCounts[2]);

    printf("D (60-69): %d\n", gradeCounts[3]);

    printf("F (<60): %d\n", gradeCounts[4]);

}
```



```
Student Grade Tracker
1. Create Course
2. Add Assignment
3. Register Student
4. Enter Grades
5. Calculate Final Grades
6. Generate Reports
0. Exit
Enter your choice: 2
Please create a course first.

Student Grade Tracker
1. Create Course
2. Add Assignment
3. Register Student
4. Enter Grades
5. Calculate Final Grades
6. Generate Reports
0. Exit
Enter your choice: 1
Enter course name: BTS
```