SET A

United International University

Department of Computer Science and Engineering

Final Examination **Spring 2024**

Course Code: **CSE 1112** Course Title: **Structured Programming Language Laboratory**

Date: May 19, 2024 Time: 09:00 AM – 10:00 AM (1 hour) Full marks: 25

Name: Student ID:

Write down C programs for the following problems in Code Blocks (or any C compiler you prefer), and present the code to your instructor after the time is up. You can make rough calculations in this paper.

Problem 1 (Marks: 13)

A word is a **Dragon Word** if it starts with '?', ends with '#' and contains only lowercase or uppercase letters in between. A string is a **Dragon String** if the middle word is a Dragon Word. Your task is to check if a given string is a Dragon String.

For this task, you have to implement the following functions: **[Cannot use string.h functions]**

1. **`int countWords(char *str)`**: Takes a string as a parameter, and counts how many words are in it. *You can safely assume that two words are separated by a space only.*
2. **`void getMiddleWord(char str[])`**: Takes the string and finds the starting index of the middle word. You can assume that there will always be odd number of words. You should use the `countWords()` function.
3. **`int isDragon(char str[])`**: Checks if the middle word is a Dragon word, if it is then the function returns 1, otherwise returns 0. You need to use `getMiddleWord()` function to find the starting index of the middle word.

In the main function, you have to take as input a string, and print if the input string is a Dragon String.

Sample input	**Sample output**	**Explanation**
`fly me to the moon`	No	The middle word 'to' is not a Dragon Word
`I love ?SpL# and UIU`	Yes	The middle word '?SpL#' is a Dragon Word
`I ?love# SpL and UIU`	No	There is a Dragon Word, but it's not the middle one

Problem 2 (Marks: 12)

You are tasked with creating a program to evaluate the "luck" of new players in a video game. Each player has allocated stat points in vitality, magic, defense, and attack. The program should determine the luck value of a player based on their stat points. First, create a structure to hold player data, including their username and stat points:

```
struct Player {
    char username[50];
    int vitality;
    int magic;
    int defense;
    int attack;
};
```

The luck value calculation involves two scenarios:

1. If the summation of the player's stat points forms a Spectacular number, the luck value is the product of the Spectacular number and 10.
2. If the summation doesn't form a Spectacular number, the luck value is just the sum of the stat points.

A Spectacular number n must meet these criteria:

- n is a prime number.
- If s is the sum of the digits of n, then $1 + 2 + 3 + \cdots + s$ is divisible by the last digit of n.

Example: 167 is a spectacular number because –

- It is a prime number.
- The sum of its digits are $1 + 6 + 7 = 14$, and $1 + 2 + 3 + \cdots + 14 = 105$, which is divisible by 7, the last digit of 167.

In this problem, you have to implement the following functions:

1. **`int isPrime(int num)`**: Checks if a given num is prime or not. Returns 1 if prime, and 0 otherwise.
2. **`int digitSum(int num)`**: Calculates the sum of digits of a given number num. **You must use recursion**.
3. **`int summation(int n)`**: Calculates the summation of numbers from 1 to num.

Sample input (bold ones are inputs from user, regular ones are prompts)	**Sample output**

`Enter player username:` **`Joey231`**
`Enter vitality:` **`47`**
`Enter magic:` **`40`**
`Enter defense:` **`40`**
`Enter attack:` **`40`**
	`167 is a spectacular number`
	`Luck value for player Joey231is: 1670`

`Enter player username:` **`Ty42`**
`Enter vitality:` **`88`**
`Enter magic:` **`89`**
`Enter defense:` **`87`**
`Enter attack:` **`34`**
	`298 is not a spectacular number`
	`Luck value for player Ty42 is: 298`

SET B

United International University

Department of Computer Science and Engineering

Final Examination **Spring 2024**

Course Code: **CSE 1112** Course Title: **Structured Programming Language Laboratory**

Date: May 19, 2024 Time: 09:00 AM – 10:00 AM (1 hour) Full marks: 25

Name: Student ID:

Write down C programs for the following problems in Code Blocks (or any C compiler you prefer), and present the code to your instructor after the time is up. You can make rough calculations in this paper.

Problem 1 (Marks: 13)

A string is a **Lucky String** if the total length of all its odd-numbered words is divisible by 7. Your task is to find out if a given string is a Lucky String or not.

For this task, you have to implement the following functions: **[Cannot use string.h functions]**

1. **`int countWords(char str[])`**: Takes a string as a parameter, and counts how many words are in it. *You can safely assume that two words are separated by a space only.*
2. **`int nNoWordLength(char* str, int n)`**: If called, it returns the length of the nth word in the string.
3. **`int luckyString(char* str)`**: If called, it returns 1 if the string is a Lucky String, and 0 otherwise. You need to use `countWords()` and `nNoWordLength()` functions to find its odd numbered words.

In the main function, you have to take as input a string, and print if the input string is a Lucky String.

Sample input	**Sample output**	**Explanation**
`fly me to the moon`	No	Length of odd-numbered words = 3 (fly) + 2(to) + 4 (moon) = 9, not divisible by 7
`I love SPL and UIU`	Yes	Length of odd-numbered words = 1 (I) + 3(SPL) + 3 (UIU) = 7, divisible by 7

Problem 2 (Marks: 12)

A school is conducting a social experiment and wants to identify students whose academic scores in four subjects — Physics, Chemistry, Biology, and Mathematics — form a special pattern. The pattern, called **Fibo-Even**, occurs when the marks obtained by students in these subjects correspond to the positions of numbers in the **Fibonacci** series, and the sum of these Fibonacci numbers is even.

For example, if a student scored 4 in Physics, 3 in Chemistry, 5 in Biology, and 6 in Mathematics (out of 40 in each subject), the positions of these marks (4th, 3rd, 5th, and 6th) in the Fibonacci series (3, 2, 5, 8) yield an even sum (3 + 2 + 5 + 8 = 18, which is even). Hence it is a Fibo-Even result.

To help with this experiment, the school needs a program that takes the names and marks of students in these subjects as input and identifies those whose scores meet the Fibo-Even criteria. Use the following structure to store student data:

```
struct Student{
    char std_name[50];
    Int result[4];
}
```

For this purpose, implement the following functions:

1. **`int Is_Even(int num)`**: This function determines whether num is an even number or not by returning 1 or 0.
2. **`int Fibonacci_Finder(int n)`**: This function finds the Fibonacci number at nth position. **You must use recursion**.
3. **`int Fibo_sum(int arr[], int size)`**: This function calculates the sum of Fibonacci numbers corresponding to the elements of the input array.

Sample input (bold ones are inputs from user, regular ones are prompts)	**Sample output**

`Name: John`
`Physics Mark: 4`
`Chemistry Mark: 3`
`Biology Mark: 5`
`Mathematics Mark: 6`
	`Sum of the Fibonacci numbers= 18`
	`John has a Fibo-Even result`

`Name: Alex`
`Physics Mark: 14`
`Chemistry Mark: 23`
`Biology Mark: 15`
`Mathematics Mark: 16`
	`Sum of the Fibonacci numbers= 30631`
	`Alex does not have a Fibo-Even result`

SET A

United International University

Department of Computer Science and Engineering

Final Examination **Spring 2024**

Course Code: **CSE 1112** Course Title: **Structured Programming Language Laboratory**

Date: May 19, 2024 Time: 11:30 AM – 12:30 AM (1 hour) Full marks: 25

Name: Student ID:

Write down C programs for the following problems in Code Blocks (or any C compiler you prefer), and present the code to your instructor after the time is up. You can make rough calculations in this paper.

Problem 1 (Marks: 12)

The Atbash cipher swaps each letter of the alphabet with its counterpart from the opposite end. For instance, 'A' becomes 'Z', 'B' becomes 'Y', and so on. Uppercase and lowercase are preserved. Special characters are replaced by spaces. For example, `"!@#Hello!@)World"` transforms into `"   Svool   Dliow"`. Your task is to write a C program that takes a string as input, applies the Atbash cipher to it and displays the output.

It must consist of the following functions: **[Cannot use string.h functions]**

1. **`void removeSpecialCharacters(char str[])`**: This function takes a pointer to a string and replaces all the special characters with a space ‘ ’.
2. **`char changeAlphabet(char alphabet)`**: The function accepts a single character and determines its counterpart in the Atbash cipher if it is an alphabetic character (either uppercase or lowercase) and returns it.
3. **`void encoder(char *p)`**: This function takes the text to encrypt. Then applying the Atbash cipher, it transforms the text into its encrypted version. Note that this function must make use of the above two functions.

In the main function, take the input string from the user and call the encoder function to encode the message and finally prints the encoded message.

Sample Input	**Sample Output**
`!@#Hello!@)World`	`   Svool   Dliow`
`ABCDEFG#hijklmnop#QRSTUV)wxyz)`	`ZYXWVUT srqponmlk JIHGFE dcba`

Problem 2 (Marks: 13)

Suppose, you are developing a ticket management system for Bangladesh Railway, where several trains are used for local transportation, and each train has specific details and limited tickets assigned to it.

Every train of Bangladesh Railway has the following details:

- Name of the train (a string)
- Total tickets (an integer)
- Ratings (a float)

You have to create a structure named `Train` to store the above details of each train. You also have to implement the following functions:

1. **`addTrain(struct Train listOfTrains[], int numOfTrains)`**: This function takes an array of Train structures and the total number of trains currently available in the system as input, and adds a new train to the system.
2. **`mostPopularTrain(struct Train listOfTrains[], int numOfTrains)`**: This function prints the name of the most popular train **using recursion**. The Most Popular Train is the one that has the highest capacity (total number of tickets). *[You are allowed to find the highest number of total tickets from this function and then, get the train name from the main function]*.
3. **`displayAllTrains(struct Train listOfTrains[], int numOfTrains)`**: This function prints the details of all the trains listed in the system.

In the main function, create an array of `Train` structure, and provide a menu for management to add trains, find the most popular train, and display all available trains in the system.

Sample Input/Output (**bold** -> user input, regular text -> console print)

```
1. Add a train
2. Most Popular Train
3. List of the trains
4. Exit

Enter your choice: 1
Name of the train: Shuborno Express
Total tickets: 150
Ratings: 4.5

1. Add a train
2. Most Popular Train
3. List of the trains
4. Exit

Enter your choice: 1
Name of the train: Mohanagar Provati
Total tickets: 200
Ratings: 4.3

1. Add a train
2. Most Popular Train
3. List of the trains
4. Exit

Enter your choice: 2
Name of the Most Popular Train:
Mohanagar Provati

1. Add a train
2. Most Popular Train
3. List of the trains
4. Exit

Enter your choice: 3
Name of the train: Shuborno Express
Total tickets: 150
Ratings: 4.5

Name of the train: Mohanagar Provati
Total tickets: 200
Ratings: 4.3
```

SET B

United International University

Department of Computer Science and Engineering

Final Examination Spring 2024

Course Code: **CSE 1112** Course Title: **Structured Programming Language Laboratory**

Date: May 19, 2024 Time: 11:30 AM – 12:30 AM (1 hour) Full marks: 25

Name: Student ID:

Write down C programs for the following problems in Code Blocks (or any C compiler you prefer), and present the code to your instructor after the time is up. You can make rough calculations in this paper.

Problem 1 (Marks: 12)

Your task is to create a **cybernetic entity detection program** that identifies and neutralizes dangerous entities. An entity is a cyber threat if the entity name is valid (check below function) and threat level is greater than 50.

It must consist of the following functions: **[Cannot use string.h functions]**

1. **`char getLastChar(char str[])`**: Takes a string as input and returns the last character of the string.
2. **`int isValidEntity(char *entityName)`**: Takes a string `entityName` representing the name of a cybernetic entity as input. Returns 1 if the entity's name meets the following criteria (and 0 otherwise):
   a. Contains only alphanumeric characters (A-Z, a-z, 0-9).
   b. Begins with an uppercase letter and end with lowercase letter (Use `getLastChar()` function to find the last character).
   c. Does not contain any special characters (@,#,$,%,^,&,*).
3. **`int isCyberThreat(char *entityName, int threatLevel)`**: Returns 1 if the `entityName` is valid (as per `isValidEntity()` function) and the threat level is greater than 50. Returns 0 otherwise.

In the main function, take as input entity name and threat level and print whether the entity is a cyber threat or not.

Sample input	Sample output

`AlphaCyber`
`30`
	`Safe`

`BetaCyber`
`80`
	`Danger !!`

Problem 2 (Marks: 13)

Suppose, you are developing a student management system for UIU, where information of all students will be stored and retrieved.

Every student in UIU has the following details:

- Name of the student (a string)
- ID (a 5-digit integer)
- CGPA (a float)

You have to create a structure named `Student` to store the above details of each student. You also have to implement the following functions:

1. **`addStudent(struct Student listOfStduents[], int numOfStduents)`**: This function takes an array of Student structures and the total number of students currently included in the system as input, and adds a new student to the system.
2. **`highestRankedStudent(struct Student listOfStduents[], int numOfStduents)`**: This function prints the name of the highest-ranked student **using recursion**. The Highest Ranked Student is the one who has the highest cgpa. *[You are allowed to find the highest cgpa from this function and then, get the student name from the main function.]*
3. **`displayAllStduents(struct Student listOfStduents[], int numOfStduents)`**: This function prints the details of all the students listed in the system.

In the main function, create an array of `Student` structure, and provide a menu for management to add students, find the highest-ranked student, and display all included students in the system.

<u>**Sample Input/Output**</u> (**bold** -> user input, regular text -> console print)

```
1. Add a student
2. Highest Ranked Student
3. List of the students
4. Exit

Enter your choice: 1
Name of the student: Md. Karim
ID: 01350
Cgpa: 3.77

1. Add a student
2. Highest Ranked Student
3. List of the students
4. Exit

Enter your choice: 1
Name of the student: Md. Rahim
ID: 01323
Cgpa: 3.93

1. Add a student
2. Highest Ranked Student
3. List of the students
4. Exit

Enter your choice: 2
Name of the Highest Ranked Student:
Md. Rahim

1. Add a student
2. Highest Ranked Student
3. List of the students
4. Exit

Enter your choice: 3
Name of the student: Md. Karim
ID: 01350
Cgpa: 3.77

Name of the student: Md. Rahim
ID: 01323
Cgpa: 3.93
```