

United International University

Department of Computer Science and Engineering

Final Examination Spring 2024Course 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 |
| ABCDEFGH#ijklmnop#QRSTUWxyz) | 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 | Name of the Most Popular Train: |
| 2. Most Popular Train | Mohanagar Provati |
| 3. List of the trains | |
| 4. Exit | 1. Add a train |
| | 2. Most Popular Train |
| Enter your choice: 1 | 3. List of the trains |
| Name of the train: Shuborno Express | 4. Exit |
| Total tickets: 150 | |
| Ratings: 4.5 | Enter your choice: 3 |
| | Name of the train: Shuborno Express |
| 1. Add a train | Total tickets: 150 |
| 2. Most Popular Train | Ratings: 4.5 |
| 3. List of the trains | |
| 4. Exit | Name of the train: Mohanagar Provati |
| | Total tickets: 200 |
| Enter your choice: 1 | Ratings: 4.3 |
| 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 | |



United International University

Department of Computer Science and Engineering

CSE 1112: Structured Programming Language Laboratory

Trimester: Summer 2024

Final Examination, Total Marks: 25, Total Time: 1 Hour

Any examinee found adopting unfair means will be expelled from the trimester/program as per UIU disciplinary rules.

Write the following C programs using a C compiler (e.g., Code::Blocks) within the given time. After completion, present the code to your examiner for evaluation.

| | |
|-------|-------------|
| Name: | Student ID: |
|-------|-------------|

Problem 1 [Marks: 13] - Hospital Patient Management System

You are required to develop a **C program** that manages patient records in a hospital. Each patient has specific details, including their name, age, gender, ID, and a condition description. The program should allow users to perform the following operations using a menu-driven approach:

- **Add a New Patient:** This allows the user to enter the details of a new patient and store them in the hospital records.
- **Search for Patients by ID:** This allows the user to search for a patient by their ID and display their details.
- **Update Patient Condition by ID:** This allows the user to update a patient's condition by providing their ID.
- **Display the List of All Patients:** This displays the details of all patients currently registered, with recovered patients listed first.
- **Exit the Program:** This terminates the program.

You must first create a structure named `Patient` to store the following details:

- Name (String)
- Age (Integer)
- Gender (Character: 'M' for male, 'F' for female)
- ID (Integer)
- Condition (String)

The program should use an array of `Patient` to store up to **100 patients**.

Instructions:

- Implement the menu using conditional statements (if-else or switch-case).
- Ensure that the maximum number of patients that can be recorded is 100.

Sample Input/Output:

Hospital Patient Management System:

1. Add a new patient
2. Search for a patient by ID
3. Update a patient's condition
4. Display all patients
5. Exit

Enter your choice: 1

Enter patient name: John Doe

Enter age: 45

Enter gender (M/F): M

Enter ID: 1002

Enter condition: Flu

Output: The patient added successfully.

Enter your choice: 2

Output: Enter patient ID to search: 1002

Patient Records:

ID: 1002, Name: John Doe, Age: 45, Gender: M, Condition: Flu

Enter your choice: 3

Enter patient ID to update: 1002

Enter new condition: Recovered

Output: Condition updated successfully.

Enter your choice: 4

Output:

Patient Records:

ID: 1002, Name: John Doe, Age: 45, Gender: M, Condition: Recovered

Enter your choice: 5

Output: Exiting the program.

Problem 2 [Marks: 12] - Spell Sentence Determination

A sentence is considered a **Spell** if it meets the following conditions:

1. No word in the sentence has more than 9 letters.
2. If you concatenate the lengths of all the words as digits, the resulting number is a prime number.

Your task is to determine if a given string is a Spell.

For this task, you must implement the following functions (Note: You are not allowed to use functions from the string.h library):

- **int number_of_words(char str[]):** This function takes a string as a parameter and returns the number of words in the string. Words are separated by spaces only.
- **int nth_word_length(char str[], int n):** This function returns the length of the nth word in the string.
- **int is_prime(int number):** This function takes an integer as a parameter and returns 1 if the number is a prime, otherwise 0.
- **int is_spell(char str[]):** This function returns 1 if the string is a Spell, otherwise 0. You should use the number_of_words(), nth_word_length(), and is_prime() functions to accomplish this.

In the main function, you must take a string as input and print whether it is a Spell or not.

Sample Input and Output:

| Sample Input | Sample Output | Explanation |
|--|---------------|---|
| Avada Kedavraaa | Yes | Two words with lengths 5 and 9, form the number 59, which is a prime. So, it is a Spell. |
| Hello World, I love C | No | Five words with lengths 5, 5, 1, 4, and 1, form the number 55141, which is divisible by 3. Hence, it is not a prime and not a Spell. |
| Every end is a new beginning in disguise | No | Eight words with lengths 5, 3, 2, 1, 3, 9, 2, and 8, form the number 53213928, which is divisible by 2. Hence, it is not a prime and not a Spell. |
| Obligatory programming course | No | The first word has 10 letters, so the string is not a Spell. |