Data Structures Project

Aim :
To create a Hangman Game using Linked List.

Data Structure Used :

The linked list is used in this Hangman game implementation to represent the state of the word that the player is trying to guess. Each node of the linked list corresponds to a character in the word. Initially, the linked list contains underscores for unguessed letters, and as the player makes correct guesses, the underscores are replaced with the correct letters.

Linked List is used in this implementation because of its features which include the following :

1. **Dynamic Length**: The length of the word to be guessed can vary, and a linked list allows for dynamic allocation and management of memory for each letter of the word.
2. **Easily Replace Letters**: Linked lists allow for easy replacement of nodes' data. As the player makes correct guesses, the guessed letters replace the underscores in the linked list.
3. **Efficient Guess Processing**: When a player guesses a letter, the program can efficiently traverse the linked list to check if the guessed letter is present in the word.
4. **Flexible Word Representation**: Linked lists allow you to represent the word in a flexible manner, allowing the game to handle words of different lengths without needing to allocate a fixed-size array.
5. **Ease of Manipulation**: Adding and modifying nodes in a linked list is straightforward, which makes it suitable for revealing correct guesses in the guessed word.
6. **Ease of Display**: Displaying the current state of the guessed word is simplified by iterating through the linked list and printing the characters.

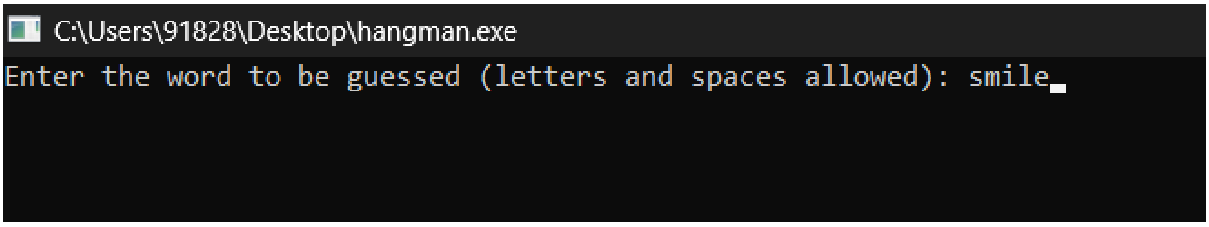
Functionality :

The Hangman game is a word-guessing game where one player thinks of a word, and another player tries to guess the word by suggesting letters. Functionality of this code:

1. Input the Word to Be Guessed
2. Initialize Guessed Word
3. Display Current State
4. User Guess Input
5. Guess Processing
6. Update Guessed Word
7. Check Win or Lose
8. Game Over Message
9. Memory Cleanup

Output :

The user has to input the word to guess.

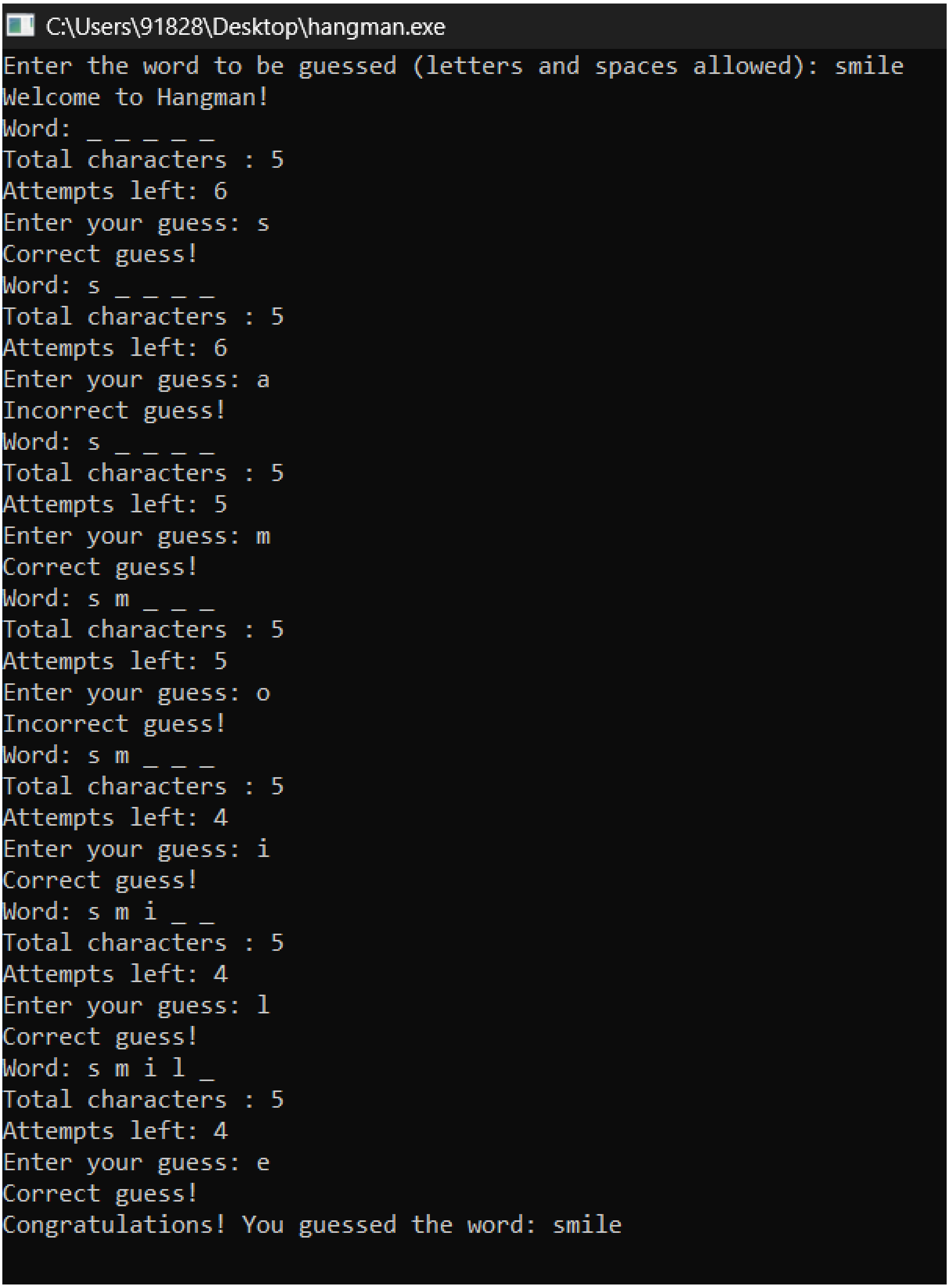


```
Enter the word to be guessed (letters and spaces allowed): smile
```

Next the user has to input their guess until they run out of attempts. Number of Attempts are reduced when the user wrongly guesses a letter and it wont be reduced for correct guesses.

When the user correctly guesses the word then the game is over and user wins.

If he is unable to correctly guess the word within the limited attempts (6), Then he loses.



```
Enter the word to be guessed (letters and spaces allowed): smile
Welcome to Hangman!
Word: _ _ _ _ _
Total characters : 5
Attempts left: 6
Enter your guess: s
Correct guess!
Word: s _ _ _ _
Total characters : 5
Attempts left: 6
Enter your guess: a
Incorrect guess!
Word: s _ _ _ _
Total characters : 5
Attempts left: 5
Enter your guess: m
Correct guess!
Word: s m _ _ _
Total characters : 5
Attempts left: 5
Enter your guess: o
Incorrect guess!
Word: s m _ _ _
Total characters : 5
Attempts left: 4
Enter your guess: i
Correct guess!
Word: s m i _ _
Total characters : 5
Attempts left: 4
Enter your guess: l
Correct guess!
Word: s m i l _
Total characters : 5
Attempts left: 4
Enter your guess: e
Correct guess!
Congratulations! You guessed the word: smile
```