1. The Library

Here is the **list of the library** that the system will use:

```
#include <stdio.h> //including the standard output
#include <stdlib.h> //including the library for memory
#include <string.h> //including the library for string modification
#include <stdbool.h> //including the library for the bool variable
```

2. Main Function

2.1 The Variable

```
int main() {
   int answer;
   trienode *root = NULL; //declaring the root as NULL
```

Lets start at the declaration first. Declaration contains the answer variable(integer) for storing the user input of the menu that the system provides and the pointer variable of the root that is set to NULL.

2.2 Menu

```
   do {
      printf("Welcome to the Slang Academy!\n");
      printf("1. Add a new slang word\n");
      printf("2. Search for a slang word\n");
      printf("3. View all words with a prefix\n");
      printf("4. View all slang words\n");
      printf("5. Exit\n");
      printf("\nSelect an option (1-5): ");

      //if statement for invalid input (string)
      if (scanf("%d", &answer) != 1){
         printf("Invalid input. Please enter a number.\n");
         while (getchar() != '\n');
         continue;
      }

      getchar();
```

The point of this code is to create a menu to guide the user about the feature available. Overall, it contains 5 different menu (adding a slang word, searching the slang word, view all the words with a prefix, and view all available words) including the exit menu. There is also an if statement for a condition that gives the user alert if the user input string type.

2.3 Menu #1

```
                  //menu for number 1
      if (answer == 1) {
         char slangSearch[54], description[101];
         int valid, i;
         //loop until the condition is met
```

```
                    do {
                      valid = 1; //check if valid

                      printf("Input a new slang word [Must be more than 1 characters and
contains no space]: ");
                      scanf(" %[^\n]", slangSearch); //ask user input

                      //condition for checking valid input
                      if (strlen(slangSearch) < 2) {
                        valid = 0;
                      } else {
                          //checking for the spaces
                        for (i = 0; slangSearch[i] != '\0'; i++) {
                          if (slangSearch[i] == ' ') {
                            valid = 0;
                            break;
                          }
                        }
                      }
                    } while (!valid);

        getchar();

        //condition to loop if the slang word description is invalid
        do {
          printf("Input a new slang word description [Must be more than 2 words]: ");
          fgets(description, sizeof(description), stdin);
          description[strcspn(description, "\n")] = '\0';
        } while (strchr(description, ' ') == NULL || strchr(strchr(description, ' ') + 1, ' ') == NULL);

        insertNode(&root, slangSearch, description); //calling the function for the insert
        waitEnter(); //calling the function to wait for the user 'enter' input
```

This is the code when the user input the number '1', it will declare a variable for the user to input the slang word and the description with two condition. If the user input doesn't meet the conditions, it will loop until both condition is fulfilled which is the slang word (Must be more than 1 characters and contains no space) and the description for the slang word (Must be more than 2 words).

When the conditions is fulfilled, the insertNode function (a function to create node)will run with the argument that user has given (can be seen on *section 4*) and the waitEnter() function (can be seen on *section 9*).

2.4 Menu #2

```
        //menu 2
    } else if (answer == 2) {
        char slangSearch[54];
        int valid, i;
        //condition check for the word
                    do {
```

```
                valid = 1; //check if valid

                printf("Input a slang word to be searched [Must be more than 1 characters
and contains no space]: ");
                scanf(" %[^\n]", slangSearch); //ask user input

                //condition for checking valid input
                if (strlen(slangSearch) < 2) {
                  valid = 0;
                } else {
                    //checking for the spaces
                  for (i = 0; slangSearch[i] != '\0'; i++) {
                    if (slangSearch[i] == ' ') {
                      valid = 0;
                      break;
                    }
                  }
                }
              } while (!valid);

    searchSlang(root, slangSearch); //calling the search function
    waitEnter(); //calling the function to wait for the user 'enter' input
```

When the user input '2'. This code will run to ask for the input of the word that the user want to search with a condition (Must be more than 1 characters and contains no space). It will loop until the condition is met and when the condition is met the search function searchSlang() will run (the search function can be seen on *section 5.2*), after that the waitEnter() function will run (the waitEnter function can be seen on *section 9*).

2.5 Menu #3

```
//menu 3
} else if (answer == 3) {
    searchprefix(root); //calling the function for prefix search
    waitEnter(); //calling the function to wait for the user 'enter' input
```

This is the code for the input '3'. The 2 function here is the searchprefix() function which you can see on *section 6.1* and also the waitEnter() function can be seen on *section 9*.

2.6 Menu #4

```
//menu 4
} else if (answer == 4) {
    printf("List of all slang words in the dictionary:\n");
    printTrie(root); //displaying all of the word in the trie
    waitEnter(); //calling the function to wait for the user 'enter' input
```

The code for input '4' will provide the list of all the slang words that the user already input. Of course, when there is no slang word the system will display an alert that there is no word that the user input, the system will call the printTrie() function that can be seen on *section 7.1* and waitEnter() function that can be seen on *section 9* .

2.7 Menu #5 and the invalid option output

```
        //menu 5
    } else if (answer == 5) {
        printf("Thank you... Have a nice day :)\n");
        freeTrie(root); //free trie memory
    } else {
        printf("Invalid option. Please choose a number between 1 and 5.\n");
    }
  } while (answer != 5);

  return 0;
}
```

this is the last code on the menu function. Which contains 2 condition, the first one is the input '5' or you can call this an exit menu. This will end the loop which will also end the system and before it end there is a freeTrie() function to free the node so it will prevent the risk of memory leak (the freetrie function can be seen on *section 8*) The Second one is the last condition that is the else condition, it will display an alert to the user and loop again until user put a valid input other than 5.

3. The Struct

3.1 Trienode struct

```
//this is the structure for the trienode including the children,description and the terminal
typedef struct trienode {
  struct trienode *children[NUM_CHAR];
  char description[NUM_CHAR];
  bool terminal;
} trienode;
```

We can say this is one of the most important part of making the trienode. Which is declaring a struct that contains the essentials, this include the struct of the children with an index for the character in the word that the user input later in the system, description of the word that the user input, and terminal that is a bool variable to give a sign of the last character in a word that is in our dictionary.

3.2 The declaration

```
//this is the function to set the struct with the data requested
trienode *createNode() {
  trienode *newNode = (trienode *)malloc(sizeof(trienode)); //setting up the newnode to allocate the memory for the newnode
  memset(newNode->children, 0, sizeof(newNode->children)); //setting up the memory size for the children declaration
  newNode->terminal = false; //setting up the default terminal to false
  newNode->description[0] = '\0'; //setting up the description to begin with nothing without words
  return newNode; //returning the node
}
```

This is to declare the all the set that we need if the user want to insert the node. For the node itself is created by allocating the memory for the node by using malloc. For the children part for the newNode is set to 0 with the sizeof the children itself this is set by using the function memset(), the terminal also is set to false which is the default, and the description of index 0 is set to '\0' (null character). The last line of the function is returning the node that we have set. Example purpose for the references to understand it, this is like when you try to set up the appearance of your character in a game.

4. The First Menu/Insert

The first menu or you can call the insert is the function that contains the code to create the trienode wether it is the root or not.

```
//function to insert trie node
bool insertNode(trienode **root, const char *text, const char *description) {
  if (*root == NULL) {
    *root = createNode(); //this is the if statement to create newnode for the root if there is no root yet in the trie
  }

  trienode *temp = *root; //declaring a temporary pointer that points to the root

  //this is a for loop to copy the letter with the iteration of an index in text
  for (int i = 0; text[i] != '\0'; i++) {
    unsigned char index = (unsigned char)text[i]; //creating a variable index to store the remaining index in the loop
    if (!temp->children[index]) {
      temp->children[index] = createNode(); //an if statement for creating a newnode if there is no node
    }
    temp = temp->children[index]; //this is for the iteration to the next node
  }

      //an if else statement for a word that has been added to the trie
  if (temp->terminal) {
    printf("You have already added this word!\n");
    return false; //returning false if the word is already exist with the sign of the terminal on the letter index
  } else {
    temp->terminal = true; //setting up the letter of the last word  to be true
    strncpy(temp->description, description, NUM_CHAR - 1); //copying the description from the inputted description
    temp->description[NUM_CHAR - 1] = '\0'; // this will save up memory for the last index
    printf("Successfully added new slang word.\n"); //output
    return true; //returning true if the word is not exist and successfully added to the trie
  }
}
```

The function process is that it will check if there is a trienode (root) in our dictionary. Best case is it will set the root to point to the newNode that has been created. Other case is the system will make a temporary pointer that points to the root and iterate while creating a nod e for each character from the word. It will also check if the word that the user has input is already in the dictionary or not. If its not then function result will run successfully by returning the true value.

5. The Search

5.1 Search Process

```
//a function to search for the node
trienode *searchNode(trienode *root, const char *word) {
   if (!root) return NULL; //returning a null if there is not root

   trienode *temp = root; //declaring temp variable

   //an iteration to store index from the word
   for (int i = 0; word[i] != '\0'; i++) {
      unsigned char index = (unsigned char)word[i]; //store the index word to the index
variable
      if (!temp->children[index]) {
         return NULL; //if there is no index that we search it will return NULL
      }
      temp = temp->children[index]; //iterate to the next index children
   }
   return temp->terminal ? temp : NULL; //return temp if the word is found but NULL if the
word is not found
}
```

The process for this function begin with the condition of returning NULL when there is no user input node. Otherwise the searching will begin by declaring a temporary pointer that point to the root and continue to iterate to search for the word that the user want to search. The iterartion will end by a condition sign on the terminal. If the terminal is true then the word user is looking for is found and NULL if the word that user is looking for is not found.

5.2 Search Result

```
//a function to searchslang words
void searchSlang(trienode *root, char *slangWord) {
   trienode *result = searchNode(root, slangWord); //call the function and store the result
in a result pointer
   if (result) { //if condition if there is a result
      printf("\nSlang word  : %s\n", slangWord);
      printf("Description : %s\n", result->description);
   } else { //a condition if there is no result
      printf("\nNo slang word \"%s\" found in the dictionary.\n", slangWord);
   }
}
```

For the result from the process before, the function begins with making a pointer that have the value of the search function (can be seen on *section 5.1*) if there is a result the system will output the slangword along with the description, else the system will output that there is no slangword the user is looking for.

6. Prefix Search

6.1 Prefix Process

```
void searchprefix(trienode *root) {
   char prefix[54]; //prefix word declaration
   printf("Input a prefix to search: ");
   scanf(" %53s", prefix);

   trienode *prefixNode = searchNode(root, prefix); //storing the searchnode function into
pointer

   //if else statement for the founded pointer
   if (!prefixNode) {
      printf("\nNo words found starting with \"%s\".\n", prefix);
   } else {
      printf("\nWords starting with \"%s\":\n", prefix);
      char buffer[100];
      strncpy(buffer, prefix, sizeof(buffer) - 1); //copying the prefix word to the buffer
      buffer[sizeof(buffer) - 1] = '\0';
      int count = 1;
      collectword(prefixNode, buffer, strlen(prefix), &count); //recursion that store the
argument of the count, buffer, and other stuff
   }
}
```

The function begin with the process of asking the user input of prefix word that the user want to search, after that it will search the word by declaring a pointer with the value of searchNode() function (can be seen on *section 5.1*) after that it process to the next condition to check if the prefixnode that the user want to search is available, not found condition that ends the function and give a status display and found condition that will lead to the print function collectword() that can be seen on *section 6.2*.

6.2 Displaying the collected words

```
// a function to collect word for the prefix print
void collectword(trienode *node, char *buffer, int len, int *count) {

        //if function for printing word
   if (node->terminal) {
      buffer[len] = '\0'; //emptying the last index of the word
      printf("%d. %s\n", (*count)++, buffer); //printing the word
   }

   //iteration for the word recursion
   for (int i = 0; i < NUM_CHAR; i++) {
```

```
        //iteration while recursion trough the children index (if statement if the index is available)
    if (node->children[i]) {
        buffer[len] = i; // making the last word
        collectword(node->children[i], buffer, len + 1, count); //recursion for the collected word
    }
  }
}
```

The function purpose is to print the word that has been found. It starts with displaying the word that has the terminal value that is true. After that it will iterate again trough the dictionary to find the first character of a word and make a recursive call to display it in the system.

7. Displaying the dictionary

7.1 Normal condition

```
//a function to display the trie
void printTrie(trienode *root) {
        //an if statement for no root
  if (!root) {
    printf("There are no slang words in the dictionary yet.\n"); //displaying
    return; //return if there is no root yet
  }

  int count = 1; //declaring the count variable
  char buffer[101]; //declaring the buffer for the word
  printTrie_rec(root, buffer, &count, 0); //calling the recursion print function
}
```

The first condition of the printing function is when there is no words available in the dictionary, it will return to the menu and give the user an alert. Second condition is when there is words in the dictionary. It begins by declaring the count and the buffer for the recursion function in *section 7.2*.

7.2 Recursive condition

```
//a function for the recursion print of the trie
void printTrie_rec(trienode *node, char *prefix, int *count, int length) {
        //an if statement for the displaying the founded word
  if (node->terminal) {
    prefix[length] = '\0'; //setting the last index to \0
    printf("%d. %s\n", (*count)++, prefix); //print the word
  }
        //iteration function to find the bool terminal that have true value
  for (int i = 0; i < NUM_CHAR; i++) {
        //the if statement if there is an index
    if (node->children[i]) {
      prefix[length] = i; //saving the first index of a word
      printTrie_rec(node->children[i], prefix, count, length + 1); //recursion of the function
```

```
    }
  }
}
```

Function begins with an if condition for the word found by detecting the value of the terminal, it will display the word. Other conditions is it will begin an iteration to search the first index of a word and store it as an argument to call on the recursion, the process will repeat until there is no words left in the dictionary. The iteration used will display the output with **lexicographical order**.

8. Preventing Memory Leak

```
//function to free memory of trie
void freeTrie(trienode *root) {
  if (!root) return; //return if there is no root

  //iterate function for freeing each node
  for (int i = 0; i < NUM_CHAR; i++) {
    if (root->children[i]) {
      freeTrie(root->children[i]);
    }
  }

  free(root); //free the memory of the node
}
```

To prevent a memory leak I provide a function that begins with a first condition to return if there is no root, to iterate trough the character nodes while clearing it, and the last line of the function is to clear the root.

9. Press to enter function

```
//this is the pressing enter function
void waitEnter() {
  printf("\nPress enter to continue...\n"); //print word
  while (getchar() != '\n'); //waiting until the user input enter
}
```

the purpose of these function is to ask the user to press enter after every interaction from the menu.

10. Input and Output Test Case

10.1 Menu #1 Adding slang word (15 case)

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: datte bayou
Input a new slang word [Must be more than 1 characters and contains no space]: datebayou
Input a new slang word description [Must be more than 2 words]: it means im okay
Successfully added new slang word.

Press enter to continue...
```

This is the input for menu number 1, it checks the **condition that is written**.

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: cap
Input a new slang word description [Must be more than 2 words]: i
Input a new slang word description [Must be more than 2 words]: it means lie
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: sus
Input a new slang word description [Must be more than 2 words]: it means suspicious
Successfully added new slang word.

Press enter to continue...
```

This will be the **trienode** that is created.

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: bet
Input a new slang word description [Must be more than 2 words]: it means okay
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: tea
Input a new slang word description [Must be more than 2 words]: it means gossip
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: lit
Input a new slang word description [Must be more than 2 words]: it means amazing
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: Drip
Input a new slang word description [Must be more than 2 words]: Stylish or fashionable
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: Flex
Input a new slang word description [Must be more than 2 words]: Means show off
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: Ghosted
Input a new slang word description [Must be more than 2 words]: end a relationship :<
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: Slay
Input a new slang word description [Must be more than 2 words]: perform exceptionally well
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: Bussin
Input a new slang word description [Must be more than 2 words]: means very good
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: babblative
Input a new slang word description [Must be more than 2 words]: prone to babble or chatter
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: backspang
Input a new slang word description [Must be more than 2 words]: sudden jolt or kick
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: balatroon
Input a new slang word description [Must be more than 2 words]: latin of "to prattle"
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: baggagery
Input a new slang word description [Must be more than 2 words]: hoi polloi
Input a new slang word description [Must be more than 2 words]: hoi polloi or rabble
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: battologize
Input a new slang word description [Must be more than 2 words]: annoy someone by repeating the same thing over and over
again
Successfully added new slang word.

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 1
Input a new slang word [Must be more than 1 characters and contains no space]: cap
Input a new slang word description [Must be more than 2 words]: it means lie
You have already added this word!

Press enter to continue...
```

The output when the word is **already written** in the dictionary.

10.2 Menu #2 Searching for slang word (5 case)

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 2
Input a slang word to be searched [Must be more than 1 characters and contains no space]: sus

Slang word  : sus
Description : it means suspicious

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 2
Input a slang word to be searched [Must be more than 1 characters and contains no space]: cap

Slang word  : cap
Description : it means lie

Press enter to continue...
```

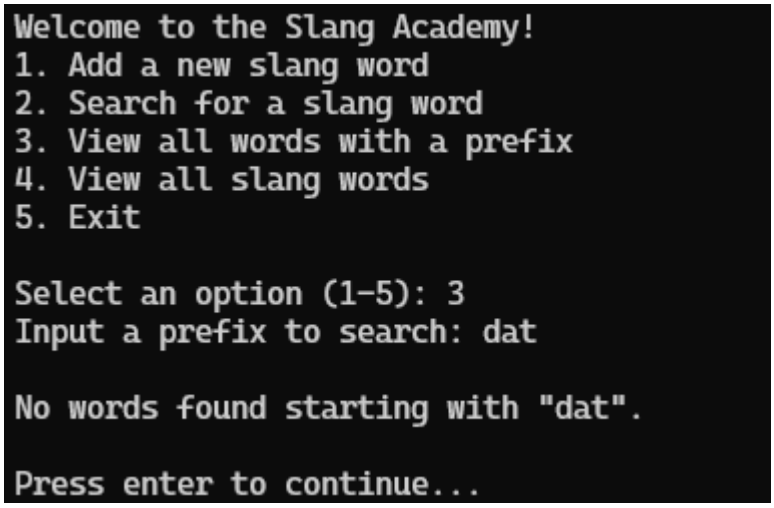
```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 2
Input a slang word to be searched [Must be more than 1 characters and contains no space]: datebayou

Slang word  : datebayou
Description : it means im okay

Press enter to continue...
```

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 2
Input a slang word to be searched [Must be more than 1 characters and contains no space]: goat

Slang word  : goat
Description : greatest of all time

Press enter to continue...
```

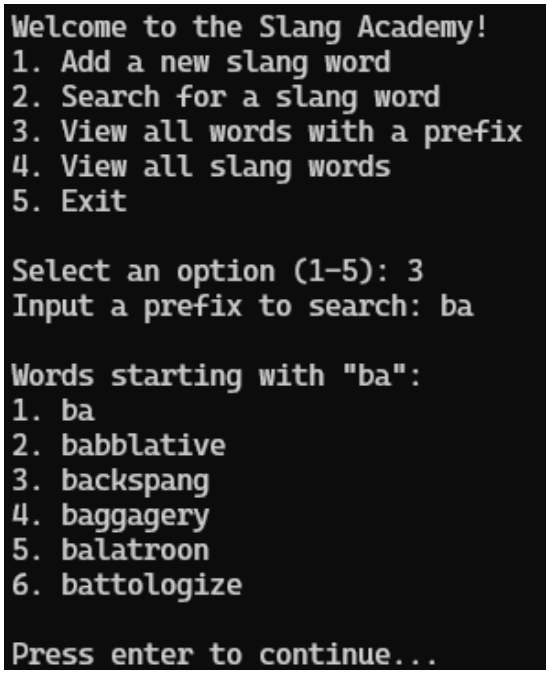
```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 2
Input a slang word to be searched [Must be more than 1 characters and contains no space]: lit

Slang word  : lit
Description : it means amazing

Press enter to continue...
```

This is the test case to **search for slang word** that has been inputted from menu 1.

10.3 Menu #3 Prefix Search

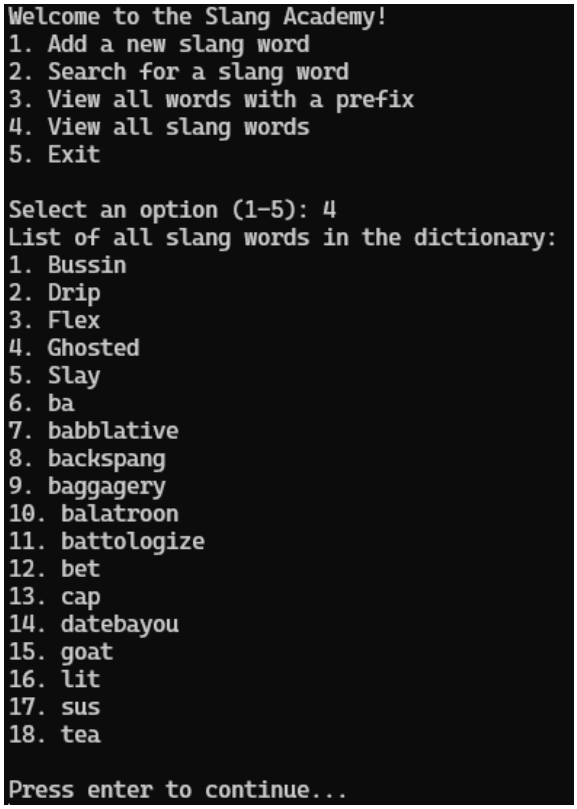


The example if the **word is not found.**

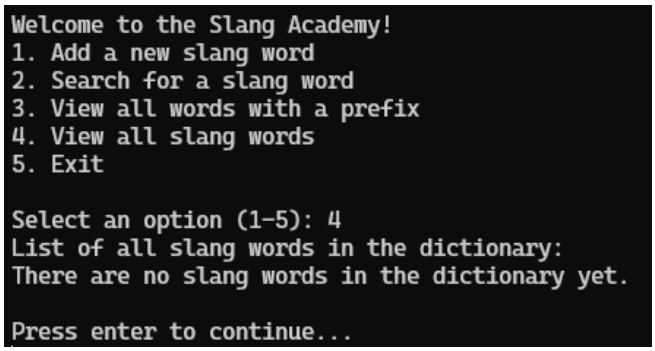


This is the example when the **word is found**.

10.4 Menu #4 View all slang words



This is the example **when there is slang word** in the dictionary. It will display in **lexicographical order.**



This is the example when there is **no words** in the dictionary.

10.5 Menu #5 Exit

```
Welcome to the Slang Academy!
1. Add a new slang word
2. Search for a slang word
3. View all words with a prefix
4. View all slang words
5. Exit

Select an option (1-5): 5
Thank you... Have a nice day :)
```

This will end the program.