

Nama : Anisa Dwi Lestari

NIM : 2702212490

Kelas : LV01

Mata Kuliah : Data Structures

Dokumentasi Program

- **Struct Data**

```
#include <stdio.h> // included for standard input output function
#include <string.h> // included for string manipulation function
#include <stdlib.h> // included for memory allocation (malloc)

// define a structure to represent node in trie
struct TrieNode{
    char ch; // character stored in the node
    char* desc; // description of the word
    int endWord; // mark the node to determine if it's the end of a word
    struct TrieNode* children[128]; // represent the children nodes of the current node
};
```

This code is used to represent a node in the trie data structure. Each node has a character, a description, a flag to show if it's the end of a word, and an array of pointers to its children nodes. I use the ASCII values of characters to index the children nodes. I take the character's ASCII value and subtract 'a' from it. It gives the perfect index to find child node in the array.

- **Create New Trie Node**

```
// function to create new trie node
struct TrieNode* createTrieNode(){
    // allocates memory for the node using malloc
    struct TrieNode* node = (struct TrieNode*)malloc(sizeof(struct TrieNode));

    // set all pointers in 'children' array to NULL
    for(int i = 0; i < 128; i++){
        node->children[i] = NULL;
    }
}
```

```

node->ch = '\0'; // set character to NULL
node->desc = NULL; // set description to NULL
node->endWord = 0; // set endWord flag to '0' or false

return node; // return to newly created node
}

```

The code creates a new node by reserving some memory with malloc. The children array is set to NULL, meaning there are no child nodes yet. The character is also set to NULL, meaning that there is no character assigned to the node. The description is also set to NULL, meaning there is no description for this node yet. The 'endWord' flag is set to '0' or false, showing that this node doesn't mark the end of a word.

- **Insert a Word and Description (Menu 1)**

```

// function to insert a word and its description into the trie
// takes 3 parameters: root node of the trie, word to be insert, description of the word
void insertTrie(struct TrieNode* root, char* word, char* desc){
    struct TrieNode* curr = root; // initialize 'curr' to the root node

    // iterates each character in the word
    for (int i = 0; word[i] != '\0'; i++){
        int idx = word[i] - 'a'; // calculates index based on ASCII value

        // if the corresponding child doesn't exist
        if(curr->children[idx] == NULL){
            // create child using 'createTrieNode' function
            curr->children[idx] = createTrieNode();
            // sets its character to the current character
            curr->children[idx]->ch = word[i];
        }

        // have a prefix match, go down a level
        curr = curr->children[idx];
    }

    curr->endWord = 1; // mark node as the end of a word

    // allocates memory for 'desc' of the node
    curr->desc = (char*)malloc(sizeof(char) * (strlen(desc) + 1));
    strcpy(curr->desc, desc); // copies description string
}

```

The code inserts a word and its description into the trie. It needs three parameters, where the trie starts (the root node), the word user wants to add, and what the description of the word. First, the function creates a 'curr' pointer that points to the root node of the trie. Then loops through each character in the word and calculates the index based on the ASCII value of the character. If the corresponding child node doesn't exist yet, the function creates a new child and sets its character to the current character. If there's a prefix match, the function moves down a level in the trie. Once all the characters in the word have been processed, the function sets 'endWord' flag to 1 to mark the current node as the end of a word. The function allocates memory for the description of the node and copies the description string into the node.

- **Release Slang Word (Menu 1)**

```
// function that allows user to add new slang word and its description
void releaseSlangWord(struct TrieNode* root){ // takes 1 parameter: root node of the trie
    char newWord[128]; // array 'newWord' to store new slang word
    int hasSpace; // tracks if the new word contains space

    // looping until user inputs at least 2 characters and no spaces
    do{
        hasSpace = 1; // set hasSpace flag to 1
        // prompt user to input new slang word
        printf("Input a new slang word [Must be more than 1 characters and contain no spaces]: ");
        scanf("%[^\n]", newWord); // read user input until a newline character
        getchar(); // clears buffer

        // iterates each character in the word
        for(int i = 0; newWord[i] != '\0'; i++){
            if(newWord[i] == ' '){ // if current character is a space
                hasSpace = 0; // set hasSpace to 0
                break; // breaks out the loop
            }
        }
    } while((strlen(newWord) < 2) || (hasSpace == 0));

    char wordDesc[128]; // array 'wordDesc' to store description
    int wordCount; // count the number of words in the description

    // looping until user inputs a description at least 3 words
    do{
        wordCount = 1; // set wordCount flag to 1
        // prompt user to input new slang word description at least 3 words
        printf("Input a new slang word description [Must be more than 2 words]: ");
        scanf("%[^\n]", wordDesc); // read user input until a newline character
        getchar(); // clears buffer
```

```

    // iterates each character in the word
    for(int i = 0; wordDesc[i] != '\0'; i++){
        if(wordDesc[i] == ' '){ // if current character is a space
            wordCount++; // count the number of words
        }
    }
}while(wordCount < 3); // looping until user inputs a description at least 3 words

// Check if the word already exists in the Trie
struct TrieNode* curr = root; // initialize 'curr' to root node

int wordExists = 1; // set wordExists to 1

// iterates each character in the word
for(int i = 0; newWord[i] != '\0'; i++){
    int idx = newWord[i] - 'a'; // calculates index based on ASCII value

    // if the corresponding child doesn't exist
    if(curr->children[idx] == NULL){
        wordExists = 0; // set wordExists to 0
        break; // breaks out the loop
    }

    // have a prefix match, go down a level
    curr = curr->children[idx];
}

// if word exists and current node marks to end of a word, update its description
if(wordExists && curr->endWord){
    free(curr->desc); // Free the previous description

    // allocates memory for 'wordDesc' of the node
    curr->desc = (char*)malloc(sizeof(char) * (strlen(wordDesc) + 1));
    strcpy(curr->desc, wordDesc); // copies description string

    // print message word update is successful
    printf("\nSuccessfully updated a slang word.\n");
}else{ // if word doesn't exists, insert into trie
    insertTrie(root, newWord, wordDesc); // insert into the trie
    // print message new word release is successful
    printf("\nSuccessfully released new slang word.\n");

    printf("Press enter to continue...\n"); // print message to prompt user to press enter
    getchar(); // wait for user to press enter
}

```

The code in function 'releaseSlangWord' adds a new slang word and its description to the trie. It needs one parameter, where the trie begins, which is the root node. First, the function asks to type a new slang word, and make sure it's at least two characters long and doesn't have any spaces. Then it creates an array newWord to store the new slang word. Next, the function asks to type a new slang word description, and make sure it's at least three words long. Then it creates an array wordDesc to store the description and counts the number of words in the description. The function checks if the word already exists in the trie. If the word exists and the current node marks the end of a word, the function updates the description of the word. If the word doesn't exist, the function inserts the word and its description into the trie.

- **Search Slang Word (Menu 2)**

```
// function that allows user to search a slang word and its description
void searchSlangWord(struct TrieNode* root){ // takes 1 parameter: root node of the trie
    char searchWord[128]; // array 'searchWord' to store the slang word
    int hasSpace; // tracks if the new word contains space

    // looping until user inputs at least 2 characters and no spaces
    do{
        hasSpace = 1; // set hasSpace flag to 1
        // prompt user to input slang word to be searched
        printf("Input a slang word to be searched [Must be more than 1 characters and
contains no space]: ");
        scanf("%[^\n]", searchWord); // read user input until a newline character
        getchar(); // clears buffer

        // iterates each character in the word
        for(int i = 0; searchWord[i] != '\0'; i++){
            if(searchWord[i] == ' '){ // if current character is a space
                hasSpace = 0; // set hasSpace to 0
                break; // breaks out the loop
            }
        }
    } while((strlen(searchWord) < 2) || (hasSpace == 0));

    // Check if the word already exists in the Trie
    struct TrieNode* curr = root; // initialize 'curr' to root node

    // iterates each character in the word
    for(int i = 0; searchWord[i] != '\0'; i++){
        int idx = searchWord[i] - 'a'; //calculates index based on ASCII value

        // if the corresponding child doesn't exist
        if(curr->children[idx] == NULL){
```

```

        // print message that word is not found
        printf("\nThere is no word \"%s\" in the dictionary.\n", searchWord);
        printf("Press enter to continue...\n"); // print message to prompt user to press enter
        getchar(); // wait for user to press enter
        return; // exit function
    }

    // have a prefix match, go down a level
    curr = curr->children[idx];
}

// if current node marks to end of a word
if(curr->endWord){
    printf("\nSlang word : %s\n", searchWord); // print slang word
    printf("Description : %s\n", curr->desc); // print description
}
printf("Press enter to continue...\n"); // print message to prompt user to press enter
getchar(); // wait for user to press enter
}

```

The code in function ‘searchSlangWord’ allows the user to search for a slang word and its description in the trie. It needs one parameter, where the trie begins, which is the root node. First, the function asks to type a slang word to be searched, and make sure it’s at least two characters long and doesn’t have any spaces. The function checks if the word exists in the trie. If the word exists, the function checks if the current node marks the end of a word. If it does, it prints the slang word and its description.

- **List Prefix (Menu 3)**

```

// function that lists all the words in the trie that have a given prefix
// takes 4 parameters: root node, prefix string, current index, count of words
void listPrefix(struct TrieNode *root, char *prefix, int x, int* count){
    struct TrieNode *curr = root; // initialize 'curr' to the root node

    if(curr->endWord){ // if current node marks to end of a word
        prefix[x] = '\0'; // set the current character to null
        printf("%d. %s\n", (*count)++, prefix); // prints the slang words
    }

    // iterates each node
    for(int i = 0; i < 128; i++){
        // if the corresponding child exists
        if(curr->children[i] != NULL){
            prefix[x] = i + 'a'; // updated the prefix array

```

```

        // call 'listPrefix' for all child of the current node
        listPrefix(curr->children[i], prefix, x + 1, count);
    }
}
}

```

The code in function 'listPrefix' lists all the words in the trie that have a given prefix. It needs four parameters, the root node of the trie, the prefix string, the current index within the prefix string, and a pointer to an integer that tracks the count of words matching the given prefix. If the current node marks the end of a word, the function sets the current character of the prefix string to null and prints the slang word. Then the function iterates through each child of the current node. If a corresponding child exists, the function updates the prefix array with the current character and recursively calls 'listPrefix' for the child node. This process continues until all child nodes have been visited.

- **View All Slang Words With Prefix (Menu 3)**

```

// function that allows user to search all the slang word that have a given trie
// takes 1 parameter: root node of trie
void viewAllSlangWordsWithPrefix(struct TrieNode *root){
    int count = 1; // initialize count words to 1
    char prefix[100]; // array 'prefix' to store the prefix input

    struct TrieNode *curr = root; // initialize 'curr' to the root node

    // prompt user to input prefix to be searched
    printf("Input a prefix to be searched: ");
    scanf("%s", prefix); // read user input
    getchar(); // clears buffer

    int len = strlen(prefix); // calculate length prefix

    for(int i = 0; i < len; i++){ // iterates each character of prefix
        int idx = prefix[i] - 'a'; // calculates index based on ASCII value

        if(curr->children[idx] == NULL){ // if the corresponding child doesn't exist
            // print message that word is not found
            printf("\nThere is no prefix \"%s\" in the dictionary.\n", prefix);
            // print message to prompt user to press enter
            printf("Press enter to continue...\n");
            getchar(); // wait for user to press enter
            return; // exit function
        }

        // have a prefix match, go down a level
    }
}

```

```

        curr = curr->children[idx];
    }

    printf("\nWords starts with '%s':\n", prefix); // print message that word is found
    listPrefix(curr, prefix, len, &count); // call 'listPrefix' function and print slang word

    printf("Press enter to continue...\n"); // print message to prompt user to press enter
    getchar(); // wait for user to press enter
}

```

The code in function ‘viewAllSlangWordsWith Prefix’ allows the user to search for all the slang words that have a given prefix. First, the function initializes the ‘count’ variable to 1 and an array ‘prefix’ to store the prefix input. It then initializes a ‘curr’ variable to the root node of the trie. Then the function prompts the user to input a prefix to be searched. It reads the user input and sets the length of the prefix. The function iterates through each character of the prefix and calculates the index based on the ASCII value. If the corresponding child node doesn’t exist, it will print a message that the prefix is not found. If the corresponding child node exists, the function sets the ‘curr’ variable to the child node and continues iterating through the characters of the prefix. Once the function reaches the end of the prefix, it prints a message that prefix is found and calls the ‘listPrefix’ function to print all the slang words that have the given prefix.

- **Print All Slang Words (Menu 4)**

```

// function that prints all the words in the trie
// takes 4 parameters: root node, array temp, current depth, count words
void printAllSlangWords(TrieNode* root, char temp[], int depth, int* count) {

    if(root->endWord){ // if current root marks to end of a word
        temp[depth] = '\0'; // set the next character of temp to null
        printf("%d. %s\n", (*count)++, temp); // prints the slang words
    }

    for(int i = 0; i < 128; i++){ // iterates each node
        if(root->children[i] != NULL){ // if the corresponding child exists
            // updates the temp array with the character of that child node
            temp[depth] = root->children[i]->ch;
            // recursive calls 'printAllSlangWords' for that child node
            printAllSlangWords(root->children[i], temp, depth + 1, count);
        }
    }
}

```


The code in function 'printAllSlangWords' prints all the words in the trie. The function checks if the current root node marks the end of a word. If it does, the function sets the next character of the 'temp' array to null and prints the slang word. Then the function iterates through each child of the current root node. If the corresponding child node exists, the function updates the 'temp' array with the character of that child node and recursively calls 'printAllSlangWords' for that child node. This process continues until all child nodes have been visited.

- **View All Slang Words (Menu 4)**

```
// function that allows user to view all the slang words in the trie
void viewAllSlangWords(TrieNode* root){ // takes 1 parameters: root node
    int count = 1; // initialize count of words to 1
    char temp[128]; // array 'temp' to store the current word

    int hasChildren = 0; // set 'hasChildren' flag to 0

    for(int i = 0; i < 128; i++){ // iterates each node
        if(root->children[i] != NULL){ // if the corresponding child exists
            hasChildren = 1; // set hasChildren to 1
            break; // breaks out the loop
        }
    }

    if(hasChildren == 0){ // if root node does not have children
        // print message that word is not found
        printf("\nThere is no slang word yet in the dictionary.\n");
    }else{ // if root node has children
        // print message that word is found
        printf("\nList of all slang words in the dictionary: \n");
        // calls 'printAllSlangWords' to print the words
        printAllSlangWords(root, temp, 0, &count);
    }

    printf("Press enter to continue...\n"); // print message to prompt user to press enter
    getchar(); // wait for user to press enter
}
```

The code in function 'viewAllSlangWords' allows the user to view all the slang words in the trie. The function initializes the 'count' variable, array 'temp', and 'hasChildren' variable to 0. The function iterates through each child of the root node. If the corresponding child node exists, sets 'hasChildren' variable to 1 and breaks out of the loop. If the root node doesn't have children, print a message that there is no slang word yet. If the root node has children, print a message that the slang words have been found and print all the slang words in trie by calling the 'printAllSlangWords' function.

- **Main Function**

```
// main function
int main(){
    struct TrieNode* root = NULL; // initialize root to NULL
    root = createTrieNode(); // call 'createTrieNode' and assign to root
    int choice; // user's choice

    // main program loop
    while(choice != 5){ // looping until user enters '5'
        system("cls"); // clear the console screen
        printf("\n");
        printf("      _____ \n");
        printf(" | _ ) / _ \\ / _ \\ / ____| | | _ _ _ \n");
        printf(" | _ \\ | | | | | | | _ | | | _ | \n");
        printf(" | | ) || | | | | | | | | | _ _ \n");
        printf(" | ____/ \\____/ \\____/ \\____| | ____ \n");

        puts("");
        // asks user to choose from 5 options
        printf("What do you want to do?\n");
        printf("1. Release a new slang word\n");
        printf("2. Search a slang word\n");
        printf("3. View all slang words starting with a certain prefix word\n");
        printf("4. View all slang words\n");
        printf("5. Exit\n");
        printf(">> ");

        scanf("%d", &choice); getchar(); // read user's choice

        switch(choice){
            case 1: // 1. Release a new slang word
                releaseSlangWord(root); // calls 'releaseSlangWord' function
                break; // breaks out the loop
            case 2: // 2. Search a slang word
                searchSlangWord(root); // calls 'searchSlangWord' function
                break; // breaks out the loop
            case 3: // 3. View all slang words starting with a certain prefix word
                // calls 'viewAllSlangWordsWithPrefix' function
                viewAllSlangWordsWithPrefix(root);
                break; // breaks out the loop
            case 4: // 4. View all slang words
                viewAllSlangWords(root); // calls 'viewAllSlangWords' function
                break; // breaks out the loop
            case 5: // 5. Exit
                printf("\nThank you... Have a nice day :)\n"); // print a message
                break; // breaks out the loop
        }
    }
}
```

```
    }  
}  
  
printf("\nMade by Anisa Dwi Lestari - 2702212490\n");  
printf("Thank you for using the program. Exiting...\n");  
  
return 0; // exit program  
}
```

The code in the main function displays the menu and waits for the user's input. The function initializes the root node of the trie to null and creates a new root node using the 'createTrieNode' function. If a user wants to release a new slang word, the user enters '1', the main function calls the 'releaseSlangWord' function and waits for the user's input, then the function returns to the main function. If a user wants to search for the slang word in the trie, user enters '2' and the function returns the result to the main function. If a user wants to search for all the slang words that have a given prefix, user enters '3' and the function displays all the slang words in the trie that start with the given prefix. If a user wants to view all the slang words in the trie, user enters '4', and the function prints all the slang words in the trie. If the user wants to exit the program, user enters '5', then the function prints a message and exits the program.

Custom Cases

- **Input of 15 slang words,**

No	Word	Description
1	chill	Same meaning as “relax”
2	cringe	Same meaning as “really embarrassing”
3	cool	Same meaning as “great”
4	cheugy	Same meaning as “out of style”
5	simp	Person who does too much for who he/she likes
6	steez	Same meaning as “effortless style”
7	swole	Same meaning as “very muscular”
8	salty	Jealous (of someone)
9	snatched	nice and cool
10	sus	Same meaning as “suspect/suspicious”
11	dude	Same meaning as “a guy”
12	drip	Same meaning as “fashionable”
13	dank	Same meaning as “excellent”
14	fam	Same meaning as “family”
15	fleek	Really good, stylish

BOOGIE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: chill

Input a new slang word description [Must be more than 2 words]: Same meaning as "relax"

Successfully released new slang word.

Press enter to continue...

BOOGIE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: cringe

Input a new slang word description [Must be more than 2 words]: Same meaning as "really embarrassing"

Successfully released new slang word.

Press enter to continue...

BOOGIE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: cool

Input a new slang word description [Must be more than 2 words]: Same meaning as "great"

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: cheugy

Input a new slang word description [Must be more than 2 words]: Same meaning as "out of style"

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: simp

Input a new slang word description [Must be more than 2 words]: Person who does too much for who he/she likes

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: steez

Input a new slang word description [Must be more than 2 words]: Same meaning as "effortless style"

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: swole

Input a new slang word description [Must be more than 2 words]: Same meaning as "very muscular"

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: salty

Input a new slang word description [Must be more than 2 words]: Jealous (of someone)

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: snatched

Input a new slang word description [Must be more than 2 words]: nice and cool

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: sus

Input a new slang word description [Must be more than 2 words]: Same meaning as "suspect/suspicious"

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: dude

Input a new slang word description [Must be more than 2 words]: Same meaning as "a guy"

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: drip

Input a new slang word description [Must be more than 2 words]: Same meaning as "fashionable"

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: dank

Input a new slang word description [Must be more than 2 words]: Same meaning as "excellent"

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: fam

Input a new slang word description [Must be more than 2 words]: Same meaning as "family"

Successfully released new slang word.

Press enter to continue...

BOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 1

Input a new slang word [Must be more than 1 characters and contain no spaces]: fleek

Input a new slang word description [Must be more than 2 words]: Really good, stylish

Successfully released new slang word.

Press enter to continue...

- **Search 5 words,**

1. chill
2. sus
3. fam
4. simp
5. cool



```
What do you want to do?
1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit
>> 2
Input a slang word to be searched [Must be more than 1 characters and contains no space]: chill

Slang word : chill
Description : Same meaning as "relax"
Press enter to continue...
|
```



```
What do you want to do?
1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit
>> 2
Input a slang word to be searched [Must be more than 1 characters and contains no space]: sus

Slang word : sus
Description : Same meaning as "suspect/suspicious"
Press enter to continue...
|
```



```
What do you want to do?
1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit
>> 2
Input a slang word to be searched [Must be more than 1 characters and contains no space]: fam

Slang word : fam
Description : Same meaning as "family"
Press enter to continue...
|
```

GOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 2

Input a slang word to be searched [Must be more than 1 characters and contains no space]: simp

Slang word : simp

Description : Person who does too much for who he/she likes

Press enter to continue...

GOOGLE

What do you want to do?

1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit

>> 2

Input a slang word to be searched [Must be more than 1 characters and contains no space]: cool

Slang word : cool

Description : Same meaning as "great"

Press enter to continue...

- **View prefix 5 words**

1. Prefix “c”
2. Prefix “ch”
3. Prefix “su”
4. Prefix “da”
5. Prefix “f”

```
BOOGLIE
```

```
What do you want to do?
1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit
>> 3
Input a prefix to be searched: c

Words starts with 'c':
1. cheugy
2. chill
3. cool
4. cringe
Press enter to continue...
|
```

```
BOOGLIE
```

```
What do you want to do?
1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit
>> 3
Input a prefix to be searched: ch

Words starts with 'ch':
1. cheugy
2. chill
Press enter to continue...
|
```

```
BOOGLIE
```

```
What do you want to do?
1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit
>> 3
Input a prefix to be searched: su

Words starts with 'su':
1. sus
Press enter to continue...
|
```

BOOGLÉ

```
What do you want to do?
1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit
>> 3
Input a prefix to be searched: da

Words starts with 'da':
1. dank
Press enter to continue...
|
```

BOOGLÉ

```
What do you want to do?
1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit
>> 3
Input a prefix to be searched: f

Words starts with 'f':
1. fam
2. fleek
Press enter to continue...
|
```

- View All

BOOGLÉ

```
What do you want to do?
1. Release a new slang word
2. Search a slang word
3. View all slang words starting with a certain prefix word
4. View all slang words
5. Exit
>> 4

List of all slang words in the dictionary:
1. cheugy
2. chill
3. cool
4. cringe
5. dank
6. drip
7. dude
8. fam
9. fleek
10. salty
11. simp
12. snatched
13. steez
14. sus
15. swole
Press enter to continue...
|
```