

Nama : Raphael Brian Pratama

NIM : 2702275024

```
#include <stdio.h> // Library untuk standard input output
#include <stdlib.h> // Library untuk memori alokasi
#include <string.h> // Library untuk memanipulasi string
#include <ctype.h> // Library untuk memanipulasi character

// Membuat dasar struct untuk trie
struct trie{
    int check; // Untuk mengecek apakah ini suatu kata atau bukan (1 = kata, 0 =
    bukan kata)
    char desc[1000]; // Untuk menampung deskripsi kata
    struct trie *edge[26]; // Untuk membuat edge yang menampung semua huruf ('a' -
    'z')
};

// Membuat function tipe struct untuk membentuk node baru
struct trie *createNewNode(){
    struct trie *newNode = (struct trie*)malloc(sizeof(struct trie)); //
    Mememorialokasikan node yang baru
    newNode->check = 0; // Mengatur check menjadi 0 karena masih tidak ada kata
    yang tersimpan
    for(int i = 0; i < 26; i++){
        newNode->edge[i] = NULL;
    }
    // For loop digunakan untuk mengeset semua edge dari 0 ke 25 menjadi NULL
    return newNode;
}

// Function untuk memasukkan string ke dalam trie
void insertNode(struct trie *root, char *word, char *desc){ // word = kata yang
    diinsert, desc = deskripsi kata tersebut
    struct trie *curr = root;
    for(int i = 0; i < strlen(word); i++){
        if(curr->edge[word[i] - 'a'] == NULL){
            curr->edge[word[i] - 'a'] = createNewNode();
        }
        curr = curr->edge[word[i] - 'a'];
    }
    strcpy(curr->desc, word);
    curr->check = 1;
}

/* Cara kerja dari function insertNode:
1. Membuat node sementara bernama curr yang value sama dengan root
2. Membuat for loop dengan i yang valuenya mulai dari 0 ke panjangnya word -
1. Jika edge dengan indexnya word index i - 'a' dari curr adalah NULL
```

(dikurang 'a' untuk meletakkan di mana huruf itu berada (a = 0, b = 1, ...)) artinya tidak ada edge dengan huruf tersebut. Jadi akan membentuk node baru. Setelah dari if, curr digantikan dengan edge yang huruf tadi.

3. Setelah selesai for loop, artinya semua huruf dari word sudah dimasukkan ke dalam trie, deskripsi akan dimasukkan ke dalam curr (edge paling akhir) dan check berubah jadi 1. (karena sudah menjadi kata) */

// Function untuk mencari suatu kata dengan tipe function bool

```
bool searchWord(struct trie *root, char *findWord){ // findWord = kata yang ingin dicari
```

```
    struct trie *curr = root;
    for(int i = 0; i < strlen(findWord); i++){
        if(curr->edge[findWord[i] - 'a'] == NULL){
            return false;
        }
        curr = curr->edge[findWord[i] - 'a'];
    }
    if(curr->check == 0) return false;
    else return true;
```

```
}
```

/* Cara kerja dari function searchWord:

1. Membuat node sementara bernama curr yang value sama dengan root
2. Membuat for loop dengan i yang valuenya mulai dari 0 ke panjangnya findWord - 1. Jika edge dengan index karakter findWord ke i dari curr adalah NULL, artinya huruf tersebut tidak ada di edge, jadi, function akan mengembalikan nilai false karena tidak ada kata tersebut di function. Jika if tidak terjadi, maka curr akan digantikan dengan edge index karakter findWord ke i.
3. Jika ternyata telah sampai di akhir kata (findWord index terakhir) dan check = 0, artinya tidak ada kata yang terbuat, function akan mengembalikan nilai false. Jika sebaliknya, function akan mengembalikan nilai true */

// Function untuk mencari suatu prefix dengan tipe function bool

```
bool searchPrefix(struct trie *root, char *prefix){ // prefix = prefix yang ingin dicari
```

```
    struct trie *curr = root;
    for(int i = 0; i < strlen(prefix); i++){
        if(curr->edge[prefix[i] - 'a'] == NULL){
            return false;
        }
        curr = curr->edge[prefix[i] - 'a'];
    }
    return true;
```

```
}
```

/* Cara kerja dari function searchPrefix:

1. Membuat node sementara bernama curr yang valuenya sama dengan root
2. Membuat for loop dengan i yang valuenya dari 0 hingga panjang prefix - 1. Jika edge dengan index prefix ke I dari curr bernilai NULL, artinya edge

dengan index karakter ke i dari prefix tidak terbuat, yang artinya tidak ada prefix tersebut. Maka, function akan mengembalikan nilai false. Jika if tidak dijalankan, maka curr akan digantikan dengan edge dengan index prefix ke i.

3. Jika loop sudah selesai, function akan mengembalikan nilai true */

```
// Function untuk mengeprint semua kata yang mengandung suatu prefix yang dicari
// prefix = Prefix yang ingin dicari, hold = string sementara untuk mengeprint
kata-katanya, idx = untuk mengetahui index ke berapa huruf di katanya.
```

```
void printAllPrefix(struct trie *root, char *prefix, char *hold, int idx){
    if (root->check == 1) {
        hold[idx] = '\0';
        printf("- %s%s\n", prefix, hold);
    }
    for (int i = 0; i < 26; i++) {
        if (root->edge[i]) {
            hold[idx] = i + 'a';
            printAllPrefix(root->edge[i], prefix, hold, idx + 1);
        }
    }
}
```

/* Cara kerja function printAllPrefix:

1. Function printAllPrefix akan berjalan secara rekursi, maka dibuatlah Batasan menggunakan if. Jika check adalah 1, artinya sudah ketemu kata. Maka, function akan mengeprint prefix + hold untuk membentuk kata yang mengandung prefix
2. For loop digunakan untuk mencari edge yang mengandung dari "a" hingga "z" (0 - 25). Jika ada edge yang mengandung karakter, maka hold index ke idx akan menyimpan karakter tersebut (setiap idx dimulai dari 0). Setelah itu, function melakukan rekursi dengan root digantikan dengan edge ke i dan idx ditambahkan satu untuk melanjutkan indexnya. */

```
// Function untuk mengeprint semua kata yang ada di trie
```

```
// hold = string sementara untuk menyimpan karakter, idx = untuk mengetahui index
ke berapa huruf di katanya
```

```
void printWord(struct trie *root, char *hold, int idx){
    if(root->check == 1){
        hold[idx] = '\0';
        printf("- %s\n", hold);
    }
    for(int i = 0; i < 26; i++){
        if(root->edge[i]){
            hold[idx] = 'a' + i;
            printWord(root->edge[i], hold, idx + 1);
        }
    }
}
```

```
/* Cara kerja function printWord kurang lebih sama dengan function printAllPrefix, bedanya di parameter tidak ada prefix dan ketika mengeprint hanya mengeprint hold (di function printAllPrefix mengeprint prefix + hold).*/
```

```
// Function untuk mengecek apakah word ada spasi atau tidak.
```

```
// word = kata yang ingin dicari
```

```
bool checkSpace(char *word){  
    for(int i = 0; i < strlen(word); i++){  
        if(word[i] == ' ') return false;  
    }  
    return true;  
}
```

```
/* Cara kerja function checkSpace adalah pertama, membuat for loop dengan i dari 0 hingga panjangnya word - 1. Jika di word index ke I ada spasi, function akan mengembalikan nilai false. Jika for loop telah selesai, artinya tidak ada spasi, function akan mengembalikan nilai true. */
```

```
// Function untuk mengecek apakah word memiliki dua kata atau tidak.
```

```
// word = kata yang ingin dicari
```

```
bool checkTwoWord(char *word){  
    for(int i = 1; i < (strlen(word) - 1); i++){  
        if(word[i] == ' ' && isalpha(word[i - 1]) != 0 && isalpha(word[i + 1]) != 0) return true;  
    }  
    return false;  
}
```

```
/* Cara kerja dari function checkTwoWord kurang lebih sama dengan function checkSpace. Bedanya adalah mengecek jika word dengan index ke i adalah spasi, index sebelumnya (i - 1) adalah sebuah alfabet dan index setelahnya (i + 1) adalah sebuah alfabet, maka function akan mengembalikan nilai true. Untuk mengecek apakah karakter di string adalah alfabet, function menggunakan isalpha dari library ctype. Jika loop telah selesai, maka tidak ada dua kata, function akan mengembalikan nilai false */
```

```
// Function untuk mendisplay menu insert dan memasukkan word ke dalam trie
```

```
void insertSlangPage(struct trie *root){
```

```
    char slang[100], desc[1000];
```

```
    do{
```

```
        printf("Input a new slang word [Must be more than 1 characters and contains no space]: ");
```

```
        scanf("%[^\n]", slang); getchar();
```

```
    }while(strlen(slang) <= 1 || !checkSpace(slang)); // Jika panjang karakter kurang dari atau sama dengan satu atau function checkSpace menghasilkan false, maka loop akan mengulang
```

```
    do{
```

```
        printf("Input a new slang word description [Must be more than 2 words]:");
```

```

        scanf("%[^\\n]", desc); getchar();
    }while(!checkTwoWord(desc)); // Jika function checkTwoWord menghasilkan nilai
false, loop akan mengulang

```

```

    if(searchWord(root, slang)){
        insertNode(root, slang, desc);
        puts("\\nSuccessfully updated a slang word.");
    }
    else{
        insertNode(root, slang, desc);
        puts("\\nSuccessfully released new slang word.");
    }

```

// Jika function searchWord menghasilkan nilai true, artinya word tersebut sudah ada sebelum dimasukkan, jadinya akan mengupdate deskripsi dan mengeprintkan "Successfully updated a slang word.". Jika searchWord false, artinya word tidak ada, slang akan dimasukkan kedalam trie dan mengeprint "Successfully released new slang word."

```

        puts("Press enter to continue...");
        getchar();
    }

```

// Function untuk mendisplay page search

```

void searchSlangPage(struct trie *root){

```

```

    char slang[100];

```

```

    do{
        printf("Input a slang word to be searched [Must be more than 1 characters
and contains no space:");
        scanf("%s", slang); getchar();
    }while(strlen(slang) <= 1 || !checkSpace(slang)); // Jika panjang slang kurang
dari sama dengan 1 atau function checkSpace menghasilkan false, loop akan
mengulang dan meminta input baru

```

```

    if(!searchWord(root, slang)){
        printf("\\nThere is no word \"%s\" in the dictionary.\\n", slang);
    }
    else{
        struct trie *curr = root; // membuat trie bernama curr bernilai root
        for(int i = 0; i < strlen(slang); i++){
            curr = curr->edge[slang[i] - 'a'];
        }

```

// for loop digunakan untuk memindahkan curr ke paling akhir kata, sehingga bisa mengakses deskripsi dari kata tersebut

```

        printf("Slang word   : %s\\n", slang);
        printf("Description  : %s\\n", curr->desc);
    }

```

```
// Jika searchWord menghasilkan nilai false, artinya tidak ada kata tersebut di dictionary, jika sebaliknya maka function akan mengeprint slang word dan deskripsinya
```

```
    puts("Press enter to continue...");  
    getchar();  
}
```

```
// Function untuk mendisplay page search prefix
```

```
void searchPrefixPage(struct trie *root){  
  
    char prefix[100], word[100];  
  
    printf("Input a prefix to be searched: ");  
    scanf("%s", prefix); getchar();  
    if(!searchPrefix(root, prefix)){  
        printf("There is no prefix \"%s\" in the dictionary.", prefix);  
    }  
    else{  
        struct trie *curr = root;  
        printf("Words starts with \"%s\":\n", prefix);  
        for(int i = 0; i < strlen(prefix); i++){  
            curr = curr->edge[prefix[i] - 'a'];  
        }  
        printAllPrefix(curr, prefix, word, 0);  
    }  
}
```

```
// Jika searchPrefix menghasilkan nilai false, artinya tidak ada prefix tersebut di dictionary, jika sebaliknya maka function akan mengeprint semua kata yang mengandung prefix tersebut
```

```
    puts("\nPress enter to continue...");  
    getchar();  
}
```

```
// Function untuk mengecek apakah slang ada atau tidak
```

```
bool checkNoSlang(struct trie *root){  
    for(int i = 0; i < 26; i++){  
        if(root->edge[i] != NULL){  
            return false;  
        }  
    }  
    return true;  
}
```

```
// Jika edge index ke i tidak bernilai NULL, artinya trie memiliki slang, function akan menghasilkan nilai false. Setelah loop selesai, artinya tidak ada slang di trie, function akan menghasilkan nilai true
```

```
}
```

// Function untuk halaman print all

```
void printAllPage(struct trie *root){  
    if(checkNoSlang(root)){  
        printf("There is no slang word yet in the dictionary.");  
    }  
    else{  
        char word[100];  
        puts("List of all slang words in the dictionary:");  
        printWord(root, word, 0);  
    }  
}
```

// Jika checkNoSlang menghasilkan nilai true, artinya tidak ada slang word. Function akan mengeprint "There is no slang word yet in the dictionary.". Jika nilai false, maka function akan mengeprint semua kata dari function printWord

```
    puts("\nPress enter to continue...");  
    getchar();  
}
```

```
int main(){
```

```
    int page;
```

```
    struct trie *root = createNewNode();
```

```
    do{
```

```
        puts("\n                BOOGLE\n");
```

```
        puts("1. Release a new slang word");
```

```
        puts("2. Search a slang word");
```

```
        puts("3. View all slang words with a certain prefix word");
```

```
        puts("4. View all slang words");
```

```
        puts("5. Exit");
```

```
        printf(">> ");
```

```
        scanf("%d", &page); getchar();
```

```
        system("cls");
```

```
        switch(page){
```

```
            case 1:
```

```
                insertSlangPage(root);
```

```
                break;
```

```
            case 2:
```

```
                searchSlangPage(root);
```

```
                break;
```

```
            case 3:
```

```
                searchPrefixPage(root);
```

```
                break;
```

```
            case 4:
```

```
                printAllPage(root);
```

```
                break;
```

```
        }
```

```
        system("cls");
```

```
    }while(page != 5);
```

```
    return 0;
}
```

TEST CASE :

```
Input a new slang word [Must be more than 1 characters and contains no space]: rizz
Input a new slang word description [Must be more than 2 words]:another word for charisma
```

```
Successfully released new slang word.
Press enter to continue...
█
```

```
Input a new slang word [Must be more than 1 characters and contains no space]: mog
Input a new slang word description [Must be more than 2 words]:when you look better than others
```

```
Successfully released new slang word.
Press enter to continue...
█
```

```
Input a new slang word [Must be more than 1 characters and contains no space]: chill
Input a new slang word description [Must be more than 2 words]:synonym for relax
```

```
Successfully released new slang word.
Press enter to continue...
█
```

```
Input a new slang word [Must be more than 1 characters and contains no space]: simp
Input a new slang word description [Must be more than 2 words]:a person who is obsessed with who he/she likes
```

```
Successfully released new slang word.
Press enter to continue...
█
```

```
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]:show off
```

```
Successfully released new slang word.
Press enter to continue...
█
```

```
Input a new slang word [Must be more than 1 characters and contains no space]: zesty
Input a new slang word description [Must be more than 2 words]:meaning you look gay
```

```
Successfully released new slang word.
Press enter to continue...
█
```

```
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]:suspect or suspicios
```

```
Successfully released new slang word.
Press enter to continue...
█
```

```
Input a new slang word [Must be more than 1 characters and contains no space]: snack
Input a new slang word description [Must be more than 2 words]:someone that is attractive
```

```
Successfully released new slang word.
Press enter to continue...
█
```


Input a new slang word [Must be more than 1 characters and contains no space]: sick
Input a new slang word description [Must be more than 2 words]:another word for cool

Successfully released new slang word.
Press enter to continue...

Input a new slang word [Must be more than 1 characters and contains no space]: whip
Input a new slang word description [Must be more than 2 words]:another word for car

Successfully released new slang word.
Press enter to continue...

Input a new slang word [Must be more than 1 characters and contains no space]: strap
Input a new slang word description [Must be more than 2 words]:another word for gun

Successfully released new slang word.
Press enter to continue...

Input a new slang word [Must be more than 1 characters and contains no space]: sigma
Input a new slang word description [Must be more than 2 words]:a person that stands above the other

Successfully released new slang word.
Press enter to continue...

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]:a lie

Successfully released new slang word.
Press enter to continue...

Input a new slang word [Must be more than 1 characters and contains no space]: nep
Input a new slang word description [Must be more than 2 words]:shorten word for nephew

Successfully released new slang word.
Press enter to continue...

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]:a way to confirm something

Successfully released new slang word.
Press enter to continue...

Input a new slang word [Must be more than 1 characters and contains no space]: unc
Input a new slang word description [Must be more than 2 words]:shorten word for uncle

Successfully released new slang word.
Press enter to continue...

List of all slang words in the dictionary:

- bet
- cap
- chill
- flex
- mog
- nep
- rizz
- sick
- sigma
- simp
- snack
- strap
- sus
- unc
- whip
- zesty

Press enter to continue...

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

Slang word : unc

Description : shorten word for uncle

Press enter to continue...

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

Slang word : nep

Description : shorten word for nephew

Press enter to continue...

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

Slang word : simp

Description : a person who is obsessed with who he/she likes

Press enter to continue...

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

Slang word : snack

Description : someone that is attractive

Press enter to continue...

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

Slang word : bet

Description : a way to confirm something

Press enter to continue...

Input a prefix to be searched: si

Words starts with "si":

- sick
- sigma
- simp

Press enter to continue...

Input a prefix to be searched: zes

Words starts with "zes":

- zesty

Press enter to continue...

█

Input a prefix to be searched: a

There is no prefix "a" in the dictionary.

Press enter to continue...

█

Input a prefix to be searched: c

Words starts with "c":

- cap
- chill

Press enter to continue...

█

Input a prefix to be searched: s

Words starts with "s":

- sick
- sigma
- simp
- snack
- strap
- sus

Press enter to continue...

█