

Tugas praktikum 😊

Nama : danendra urdha bhanu cetta harenndy

Nim : 1203230110

Kelas: iF 03-01

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_LENGTH 2024
#define MIN_LENGTH 1945

void lessThanRequired(int *length) {
    *length = MIN_LENGTH;
    printf("The length of your text is less than specified, please update your text\n");
}

void equalThanRequired() {
    printf("Thank you, Your text length is correct\n");
}

void moreThanRequired(int *length) {
    *length = MAX_LENGTH;
    printf("Your text is too long, please reduce the text\n");
}

int main() {
    int lengthOfText, selectOption;
    FILE *fptr = NULL;
    char text[MAX_LENGTH];

    fptr = fopen("file.txt", "r");

    if (fptr == NULL) {
        printf("Error opening file\n");
        exit(1);
    }

    fgets(text, MAX_LENGTH, fptr);

    fclose(fptr);

    lengthOfText = strlen(text);

    selectOption = lengthOfText < MIN_LENGTH ? 0 : (lengthOfText > MAX_LENGTH ? 2 : 1);
```

```

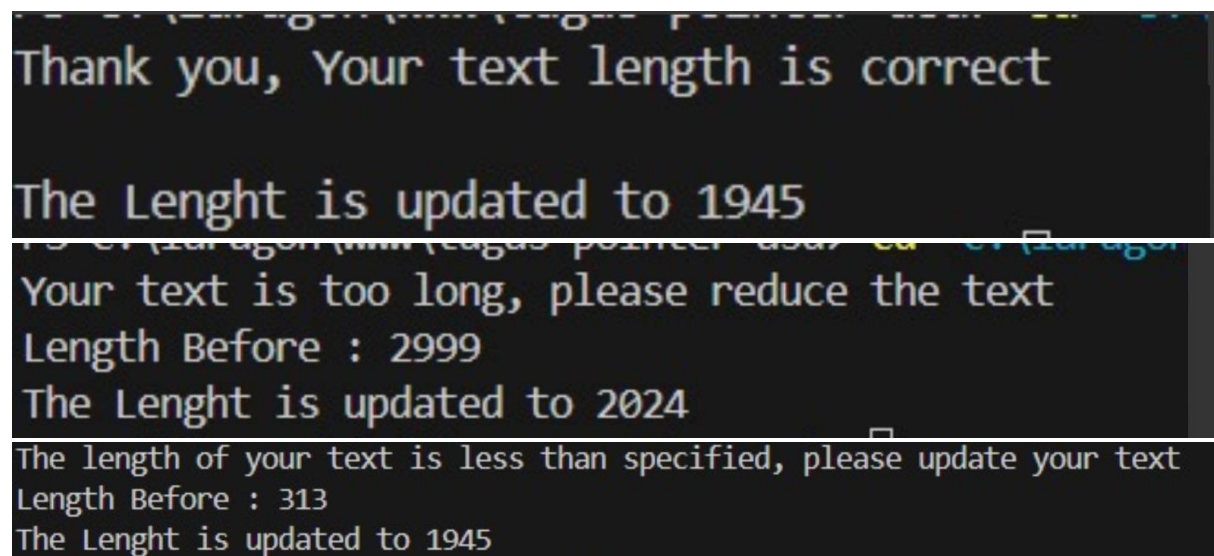
    void (*fn[])(int *) = {lessThanRequired, equalThanRequired,
moreThanRequired};
    fn[selectOption](&lengthOfText);

    printf("Length Before : %d\n", strlen(text));
    printf("The Lenght is updated to %d\n", lengthOfText);

    return 0;
}

```

Output:



```

Thank you, Your text length is correct
The Lenght is updated to 1945
Your text is too long, please reduce the text
Length Before : 2999
The Lenght is updated to 2024
The length of your text is less than specified, please update your text
Length Before : 313
The Lenght is updated to 1945

```

Penjelasan:

1. **Fungsi** checkLenghtRequirement: Mengembalikan nilai 0 jika panjang teks kurang dari MIN_LENGTH, 1 jika sama dengan MIN_LENGTH, dan 2 jika lebih dari MAX_LENGTH.

Fungsi ini menggunakan fungsi strlen untuk menghitung panjang teks. Fungsi strlen menghitung jumlah karakter dalam string yang diakhiri dengan karakter null (\0).

Setelah menghitung panjang teks, fungsi ini akan membandingkannya dengan nilai MIN_LENGTH dan MAX_LENGTH. Berikut adalah kondisi yang akan diperiksa:

Jika panjang teks kurang dari MIN_LENGTH: Fungsi ini akan mengembalikan nilai 0.

Jika panjang teks sama dengan MIN_LENGTH: Fungsi ini akan mengembalikan nilai 1.

Jika panjang teks lebih dari MAX_LENGTH: Fungsi ini akan mengembalikan nilai 2.

2. Array fn menyimpan pointer ke tiga fungsi: lessThanRequired, equalThanRequired, dan moreThanRequired. Array ini digunakan untuk memilih fungsi yang akan dipanggil berdasarkan nilai yang dikembalikan oleh fungsi checkLenghtRequirement.

Array fn dideklarasikan sebagai berikut:

```
void (*fn[])(int *) = {lessThanRequired, equalThanRequired, moreThanRequired};
```

void (*fn[]) : Deklarasi array pointer ke fungsi yang mengembalikan nilai void.

int * : Tipe data parameter yang diterima oleh fungsi-fungsi dalam array.

Array fn memiliki tiga elemen, yaitu:

fn[0]: Pointer ke fungsi lessThanRequired.

fn[1]: Pointer ke fungsi equalThanRequired.

fn[2]: Pointer ke fungsi moreThanRequired.

Fungsi yang akan dipanggil ditentukan oleh nilai yang dikembalikan oleh fungsi checkLengthRequirement. Berikut adalah contohnya:

Jika fungsi checkLengthRequirement mengembalikan nilai 0, maka fungsi lessThanRequired akan dipanggil.

Jika fungsi checkLengthRequirement mengembalikan nilai 1, maka fungsi equalThanRequired akan dipanggil.

Jika fungsi checkLengthRequirement mengembalikan nilai 2, maka fungsi moreThanRequired akan dipanggil.

Misalkan nilai selectOption adalah 1. Berikut adalah cara kerja array fn:

Kode fnselectOption: &lengthOfText akan mengevaluasi fn[1] karena nilai selectOption adalah 1.

fn[1] adalah pointer ke fungsi equalThanRequired.

Fungsi equalThanRequired akan dipanggil dengan parameter &lengthOfText.

3. **Operasi pointer:** Dalam fungsi lessThanRequired dan moreThanRequired, nilai lengthOfText diubah melalui pointer.

Operasi Pointer dalam Fungsi lessThanRequired dan moreThanRequired

Dalam fungsi lessThanRequired dan moreThanRequired, nilai lengthOfText diubah melalui pointer.

Berikut adalah penjelasan lebih detail:

Fungsi lessThanRequired:

Fungsi ini menerima satu parameter yaitu pointer ke variabel lengthOfText.

Di dalam fungsi, nilai lengthOfText diubah dengan menggunakan operator dereferencing (*).

Operator dereferencing digunakan untuk mendapatkan nilai yang disimpan di alamat memori yang ditunjuk oleh pointer.

Dalam kasus ini, nilai lengthOfText diubah menjadi nilai MIN_LENGTH.

Contoh:

Misalkan nilai lengthOfText adalah 15. Berikut adalah cara kerja operasi pointer dalam fungsi lessThanRequired:

Kode *lengthOfText = MIN_LENGTH; akan mengubah nilai yang disimpan di alamat memori yang ditunjuk oleh lengthOfText.

Karena lengthOfText adalah pointer ke variabel lengthOfText, maka nilai lengthOfText akan diubah menjadi MIN_LENGTH.

Fungsi moreThanRequired:

Fungsi moreThanRequired bekerja dengan cara yang sama dengan fungsi lessThanRequired, bedanya nilai lengthOfText diubah menjadi nilai MAX_LENGTH.

Contoh:

Misalkan nilai lengthOfText adalah 25. Berikut adalah cara kerja operasi pointer dalam fungsi moreThanRequired:

Kode *lengthOfText = MAX_LENGTH; akan mengubah nilai yang disimpan di alamat memori yang ditunjuk oleh lengthOfText.

Karena lengthOfText adalah pointer ke variabel lengthOfText, maka nilai lengthOfText akan diubah menjadi MAX_LENGTH.

