

Benediktus Daniel Afriant

1203230005

Informatika 03-03

Source Code :

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

#define MAX_LENGTH 10
#define MIN_LENGTH 7

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

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

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

int checkLenghtRequirement(char* text){
    int length = strlen(text);
    if (length < MIN_LENGTH)
        return 0;
    else if (length == MIN_LENGTH)
        return 1;
    else
        return 2;
}

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

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

    if(fptr == NULL){
```

```

        printf("Error");
        exit(1);
    }

    fgets(text, MAX_LENGTH, fptr);

    fclose(fptr);

    selectOption = checkLenghtRequirement(text);

    // TODO: Use function pointers to call the appropriate function based on
selectOption
    // HINT: Create an array of function pointers and call the appropriate
function using selectOption
    void (*functionArray[3])(int*) = {lessThanRequired, equalThanRequired,
moreThanRequired};
    functionArray[selectOption](&lengthOfText);

    printf("\nThe Length is updated to %d", lengthOfText);

    return 0;
}

```

Output :

```

PS C:\Users\Daniel\Downloads\PS1\output> cd 'c:\Users\Daniel\Downloads\PS1\output'
PS C:\Users\Daniel\Downloads\PS1\output> & .\'Test Code 44.exe'
Your text is too long, please reduce the text

The Length is updated to 7
PS C:\Users\Daniel\Downloads\PS1\output> cd 'c:\Users\Daniel\Downloads\PS1\output'
PS C:\Users\Daniel\Downloads\PS1\output> & .\'Test Code 44.exe'
The length of your text is less than specified, please update your text

The Length is updated to 7
PS C:\Users\Daniel\Downloads\PS1\output> cd 'c:\Users\Daniel\Downloads\PS1\output'
PS C:\Users\Daniel\Downloads\PS1\output> & .\'Test Code 44.exe'
Your text is too long, please reduce the text

The Length is updated to 7
PS C:\Users\Daniel\Downloads\PS1\output> cd 'c:\Users\Daniel\Downloads\PS1\output'
PS C:\Users\Daniel\Downloads\PS1\output> & .\'Test Code 44.exe'
Thank you, Your text length is correct

The Length is updated to -1892019280
PS C:\Users\Daniel\Downloads\PS1\output> 

```

Penjelasan :

1. #include <stdio.h>: Menggunakan preprocessor directive untuk menyertakan file header standar stdio.h, yang berisi deklarasi fungsi input-output standar dalam bahasa C.
2. #include <stdlib.h>: Preprocessor directive untuk menyertakan file header standar stdlib.h, yang berisi deklarasi fungsi-fungsi standar untuk alokasi memori, konversi tipe data,

pengendalian program, dan fungsi umum lainnya.

3. `#include <string.h>`: Preprocessor directive untuk menyertakan file header standar `string.h`, yang berisi deklarasi fungsi-fungsi untuk manipulasi string.

4. `#define MAX_LENGTH 10`: Menggunakan preprocessor directive `#define` untuk mendefinisikan konstanta `MAX_LENGTH` dengan nilai 10. Konstanta ini digunakan untuk menentukan panjang maksimum dari array text.

5. `#define MIN_LENGTH 7`: Menggunakan preprocessor directive `#define` untuk mendefinisikan konstanta `MIN_LENGTH` dengan nilai 7. Konstanta ini digunakan sebagai panjang minimum yang diperlukan untuk teks.

6. `void lessThanRequired(int* lengthOfText) { ... }`: Mendefinisikan fungsi `lessThanRequired` yang mengambil pointer ke int sebagai argumennya. Fungsi ini mencetak pesan bahwa panjang teks kurang dari yang ditentukan dan mengatur panjang teks ke `MIN_LENGTH`.

7. `void equalThanRequired(int* lengthOfText) { ... }`: Mendefinisikan fungsi `equalThanRequired` yang mengambil pointer ke int sebagai argumennya. Fungsi ini mencetak pesan bahwa panjang teks sudah sesuai dan tidak perlu diubah.

8. `void moreThanRequired(int* lengthOfText) { ... }`: Mendefinisikan fungsi `moreThanRequired` yang mengambil pointer ke int sebagai argumennya. Fungsi ini mencetak pesan bahwa panjang teks terlalu panjang dan mengatur panjang teks ke `MIN_LENGTH`.

9. `int checkLengthRequirement(char* text) { ... }`: Mendefinisikan fungsi `checkLengthRequirement` yang mengambil string sebagai argumennya. Fungsi ini mengembalikan nilai sesuai dengan panjang string input.

10. `int main() { ... }`: Mendefinisikan fungsi `main`, fungsi utama yang akan dieksekusi saat program dijalankan.

11. `int lengthOfText, selectOption;`: Deklarasi dua variabel `lengthOfText` dan `selectOption` sebagai integer.

12. `FILE *fptr = NULL;`: Deklarasi pointer ke `FILE` dengan nama `fptr` dan diinisialisasi dengan `NULL`.

13. `char text[MAX_LENGTH];`: Deklarasi array text dengan panjang maksimum yang sudah ditentukan oleh `MAX_LENGTH`.

14. `fptr = fopen("file.txt", "r");`: Membuka file "file.txt" dalam mode baca ("r") dan menetapkan file pointer ke `fptr`.

15. `if(fptr == NULL) { printf("Error"); exit(1); }`: Memeriksa apakah `fptr` null, jika iya, cetak pesan kesalahan dan keluar dari program dengan status keluar 1.

16. `fgets(text, MAX_LENGTH, fptr);`: Membaca baris pertama dari file yang diarahkan oleh `fptr` dan menyimpannya di dalam array `text`. Maksimum karakter yang dibaca adalah `MAX_LENGTH - 1` karakter.

17. `fclose(fp);`: Menutup file yang diarahkan oleh `fptr`.

18. `selectOption = checkLengthRequirement(text);`: Memanggil fungsi `checkLengthRequirement` untuk memeriksa panjang teks yang dibaca dari file dan menyimpan hasilnya ke dalam `selectOption`.

19. `void (*functionArray[3])(int*) = {lessThanRequired, equalThanRequired, moreThanRequired};`: Mendeklarasikan array `functionArray` yang berisi tiga pointer ke fungsi, masing-masing bertipe `void (*)(int*)`. Pointer-pointer ini menunjuk ke tiga fungsi yang telah didefinisikan sebelumnya.

20. `functionArray[selectOption](&lengthOfText);`: Memanggil fungsi yang sesuai dari array `functionArray` berdasarkan nilai `selectOption` dan melewati alamat dari `lengthOfText` ke

fungsi tersebut.

21. `printf("\nThe Length is updated to %d", lengthOfText);`: Mencetak pesan yang menyatakan bahwa panjang telah diperbarui ke nilai yang baru, yaitu `lengthOfText`.

22. `return 0;`: Mengembalikan nilai 0, menandakan bahwa program berakhir tanpa kesalahan