**Rapor**

**Kod Kısmı:**

/\*

Huseyin Soylu - 171418012

Marmara Universitesi Bilgisayar Muhendisligi Bolumu

BLM2002-Veri Yapilari ve Algoritmalar Dersi - Odev4

Asagidaki program stack ve dosya islemleri ile alakali gelistirilmis fonksiyonlari kapsar

Program main fonksiyonuna bir adet .txt uzantili arguman almaktadir. Almis oldugu bu dosyayi

satir satir okuyarak satirlarin polindrom cumle, kelime, sayi ya da harf olup olmadigini kontrol eder.

Kodlama yapilirken macar notasyonu esas alinmis ve tum degiskenler icin asagida

belirtilen esaslarda kodlama yapilmistir.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

------For Variable-----------------

variable = v; tip sonrasi eki

-------------------------------

------Use Case-----------------

integer icin = iv;

char icin =cv;

double icin = dv;

float icin = fv;

-------------------------------

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

------For Pointer------------------

pointer = p; tip sonrasi eki

-------------------------------

------Use Case-----------------

int pointer icin = ip;

char pointer icin = cp;

-------------------------------

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

------ For Array---------------

array = a; tip sonrasi eki

-------------------------------

------Use Case-----------------

int array = ia;

char array = ca;

-------------------------------

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

------For Function-------------

function = f; tip sonrasi eki

-------------------------------

------Use Case-----------------

int geri donuslu fonksiyon = if;

char geri donuslu fonksiyon = cf;

int pointer geri donuslu fonksiyon = ipf;

char pointer geri donuslu fonksiyon = cpf;

geri donussuz fonksiyon icin = vf;

-------------------------------

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

------For Function Parameter---

parameter = p;

-------------------------------

------Use Case-----------------

int geri donuslu parametreli fonsiyon = ifp;

int pointer donuslu parametreli fonksiyon = ipfp;

Seklinde tanimlanmistir.

-------------------------------

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\*/

//--------------------------------------Program Area--------------------------------------

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<malloc.h>

#include <sys/stat.h>

//-------------------------------------Stack Declare--------------------------------------

char\* stack;

int top = -1;

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//--------------------------------Stack Functinon Declare---------------------------------

// push function

void push(char value)

{

stack[++top] = value;

}

// pop function

char pop()

{

return stack[top--];

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//--------------------------------Program Functinon Declare-------------------------------

int ifp\_is\_polindrom(char array[]){

int length = strlen(array);

// Allocating the memory for the global stack pointer

stack = (char\*)malloc(length \* sizeof(char));

// Finding the mid

int i;

int iv\_array\_mid = length / 2;

for (i = 0; i < iv\_array\_mid; i++) {

push(array[i]);

}

// Verilen dizi uzunlugu ikiye tam bolunmuyor ise i ye bir ekle .

if (length % 2 != 0) {

i++;

}

//Pop function variable

char cv\_stack\_pop;

//while loop: element by element move in array

while (array[i] != '\0') {

if(top != -1){

cv\_stack\_pop = pop();

}

else{

printf("Stack Overflow - polindrom function failed!\n");

}

//One character diffirent: Function end

if (cv\_stack\_pop != array[i])

return 0;

i++;

}

return 1;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

char \* cpfp\_remove\_array\_spaces(char \*cp\_array)

{

if(cp\_array == NULL){

printf("(!)Pointer: cp\_array \n(X)Error: Pointer is not created.\n");

exit(1);

}

int i = 0, j = 0;

while (cp\_array[i])

{

//32 ==> Space Ascii Code

if (cp\_array[i] < 32 && cp\_array[i] > 45){

cp\_array[j++] = cp\_array[i];

}

i++;

}

cp\_array[j] = '\0';

return cp\_array;

}

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void vfp\_read\_file\_line(const char \* ccp\_file\_path, FILE \* fp){

if(ccp\_file\_path == NULL){

printf("(!)Pointer: ccp\_file\_path \n(X)Error: Pointer is not created.\n");

exit(1);

}

if(fp == NULL){

printf("(!)Pointer: fp \n(X)Error: Pointer is not created.\n");

exit(1);

}

if((fp = fopen(ccp\_file\_path, "r")) != NULL){

struct stat sb;

stat(ccp\_file\_path, &sb);

//sb.st\_size --> Size of file

//printf("%ld",sb.st\_size);

char \*cp\_file\_contents = malloc(sb.st\_size);

if(cp\_file\_contents == NULL){

printf("(!)Function: vfp\_read\_file\_line \n(X)Error: cp\_file\_contents pointer is not created\n");

exit(1);

}

while (fscanf(fp, "%[^\n] ", cp\_file\_contents) != EOF) {

printf(">Is -%s- Polindrom ==> ", cp\_file\_contents);

//Function: Remove space in array

cp\_file\_contents = cpfp\_remove\_array\_spaces(cp\_file\_contents);

//Polindrom control function

if (ifp\_is\_polindrom(cp\_file\_contents)) {

printf("Yes");

}

else {

printf("No");

}

printf("\n");

}

fclose(fp);

}

else{

printf("(!)Function: vfp\_read\_file\_line \n(X)Error: File not found or open file error.\n");

}

}

//-----------------------------------Main Area-------------------------------------------

int main(int argc, char\*argv[]){

if(argc != 2){

printf("The program have an parameter. The parameter is file path.\n");

exit(1);

}

const char \* ccp\_file\_path = argv[1];

FILE \* fp = malloc(sizeof(FILE\*));

if(fp == NULL){

printf("(!)Pointer: fp \n(X)Error: Pointer is not created.\n");

exit(1);

}

vfp\_read\_file\_line(ccp\_file\_path, fp);

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n");

printf("Program kucuk buyuk harf duyarli ve noktalama isaretlerini yok saymaktadir.");

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");

return 0;

}

UML:

UML PDF şeklinde dosyanın içerisindedir.

Açıklama:

Aşağıdaki program stack ve dosya işlemleri ile alakalı geliştirilmiş fonksiyonları kapsar.

Program main fonksiyonuna bir adet “.txt” uzantılı string dosya yolunu parametre almaktadır. Almış olduğu bu dosyayı satır satır okuyarak satırların polidrom cümle, kelime, harf ya da sayı olup olmadığını kontrol eder.

Kodlama yapılırken macar notasyonu esas alınmış ve kod bloğuna başlamadan önce yorum satırlarına eklenmiştir.