Practical Exercise 2

Ομάδα : 3

Συμμετέχοντες : Μιχάλης Μιχαήλ

Σώτος Βασιλείου

Πασιουρτίδης Κώστας



Προγραμμα 1:

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

struct node {

char data;

struct node \* next;

} \*head,\*head1;

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

void displayNode(struct node \*r); //to tipoma tis listas

void addNode(char newdata); //na vazo charaktires mesa

void pop(); //eksagwgi enos stixiou apo tin lista

void destroy(); //apodesmevsi mnimis

int count=0; //global metavliti pou tin xrisimopoio gia eksagwgi komvou sti sinartisi pop

Αρχικα εχω το structure μου για την δημιουργια της λιστας που θα δουμε στη συνεχια. Μετα ακολουθουν η συναρτησεις μου που γραφω ξεκαθαρα σε comments ποια είναι η ασχολια τους. Πχ η destroy είναι για αποδεσμευση μνημης όπως ζητα η εκφωνηση όταν ο χρηστης δωσει ; κλπ.

Η μεταβλητη count την εκανα global μεταβλητη γιατι είναι μεταβλητη που την χρησιμοποιώ μονο στο function της pop ώστε να μετρα αντιστροφα (μετρητης --) και να κανω εξαγωγη κομβου.

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

struct node \*n; //dimiourgia tis listas mou

head=NULL;

bool flag\_out=true; //einai true gia na kano to loop me to menu.. otan omws einai false //logo tis '.' tote termatizi t programma

do{

printf("\n\nEnter the size of the queue\n");

scanf("%d",&N);

char array[N][P];

printf("\nSelect the type of the LIFO queue\n");

printf("1 for Array or 2 for Queue\n");

scanf("%d",&k);

char charak[N];

int i;

printf("\nGive characters\n");

scanf("%s",charak);

if (k==1){

for (i=0; i<N; i++)

array[i][P]=charak[i];

}

//pame gia queue

else if (k==2){

for (i=0; i<N; i++)

addNode(charak[i]);

}

Μετα αρχικοποιω την λιστα μου ( είναι η n ) την αδειαζω με null μετα εχω bool ένα flag εξωτερικο που με βοηθα για το loop όπως εξηγω και στα comments για να επιστρεφει στο αρχικο menu από την αρχη του do while. Μετα εχω 2 επιλογες αν το κ==1 για πινακα και κ==2 για λιστα και αποθηκευω τους χαρακτηρες με τον εξης τροπο ( για κ==2 εχουμε λιστα και η συναρτηση addNode βαζει τους χαρακτηρες στη λιστα αυτη).

char choice; //epilogi tou xristi

int rem,L=N; //rem=remove diladi posous xaraktires thelei na afairesi

bool flag=true; //einai true gia to loop otan einai false logo tou ';' tote vgainei apo to loop gia na epistrepsi sto arxiko menu kai na ksanakanei loop

while(flag==true){

printf("\nCHOICES OF MENU\n");

printf("Enter an asterisk '\*' to remove elements from the queue?\n");

printf("Enter a semicolon ';' if you want to print all the queue elements!\n");

printf("Enter a dot '.' if you want to exit!\n");

scanf(" %c",&choice);

if (choice=='\*'){

printf("\nHow many elements do you want to remove?\n");

scanf("%d",&rem);

printf("\n");

if (k==2){ //edw gia queue

for(int i=0; i<rem; i++)

pop();

printf("\n");

}

else if(k==1){ //edw gia pinaka

int j=0;

while (j<rem){

printf("%c, ",array[L-1][P]);

array[L-1][P]=NULL;

L--;

j++;

}

}

}

Τις μεταβλητες τις εξηγω σε comments. Μετα ξεκινα το while με τις κυριες επιλογες. Για κ==2 είναι λιστα για κ==1 είναι πινακας. Στο \* αφαιρουμε τα στοιχεια και η μεταβλητη είναι rem που δινει ο χρηστης (δηλαδη είναι η remove).

else if (choice==';'){ //edw gia queue

if (k==2){

printf("All the queue elements are: ");

displayNode(n); //tiponi tin lista

printf("\nThe memory is cleaned\n");

destroy(); //apodesmevsi tis listas

}

else if (k==1){ //edw gia pinaka

for (i=0; i<L; i++){

printf("%c ",array[i][P]);

array[i][P]=NULL;

}

printf("\nThe memory is cleaned\n");

}

flag=false;

}

else if(choice=='.'){

printf("\nEXIT\nTHANK YOU");

flag=false;

flag\_out=false;

break;

}

}

}while(flag\_out==true);

return 0;

}

Μετα αν η επιλογη είναι ; τοτε τυπωνει τα στοιχεια αναλογα με το κ αν είναι πινακας η λιστα εξηγω και στα comments.

Αν είναι . τοτε τερματιζει το προγραμμα εχωντας και τα 2 flag και flag\_out να είναι false ώστε να βγουν απ΄τα λοοπ και να τερματισει ο κωδικας.

Functions

void addNode(char newdata)

Η συναρτηση είναι υπευθυνη για την εισαγωφη κομβου δινοντας ως παραμετρο τους χαρακτηρες που θελω να βαλω στη λιστα.

void displayNode(struct node \*r)

Η συναρτηση είναι υπευθυνη για να τυπωνει την λιστα μου. Οσους χαρακτηρες εχω θα τους τυπωσει.

void pop()

Η συναρτηση είναι υπευθυνη για να κανει εξαγωγη κομβου από την λιστα μου.

void destroy()

Η συναρτηση είναι υπευθυνη για να αποδεσμευει από την μνημη την λιστα μου

Άσκηση 3

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

Αρχικά Ανοίγουμε το file του γράφου ( σε μορφή .graph ) και διαβάζουμε τον αριθμό των κόμβων . Στην συνέχεια δημιουργούμε δυναμικά χώρο για τους κόμβους και στην συνέχεια δημιουργούμε δυναμικά και τις στήλες του ( όλες οι πιθανές επιλογές των χωρών που μπορεί να επισκεφθεί ) . Στην συνέχεια μέχρι να τελειώσει το αρχείο διαβάζουμε τις ακμές και το ανάλογο βάρος του και τις δημιουργούμε με την ανάλογη συνάρτηση .Τέλος εκτυπώνουμε τις ακμές του γράφου .

Στην συνάρτηση με τις ακμές ελέγχουμε στο να μην βγει ο Γράφος εκτός ορίων και επίσης ελέγχουμε αν υπάρχει ήδη η ακμή ( μη κατευθυνόμενη ) .Όταν γίνουν οι πιο πάνω ενέργειες προσθέτουμε την ακμή και στις δύο χώρες που τις περιλαμβάνει .

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

struct graph {

int \*\*array;

int N; //korifes

};

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

main(){

GRAPH g; #dimiourgia tou struct grafou

graph\_init\_from\_file(&g,"graph.graph"); #arxikopoiisi grafou , diavasma arxiou , apothikeusi akmwn

ektipwsi\_graphou(g);

}

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

graph\_init\_from\_file(GRAPH , filename){

fo=fopen(filename , "r");

fscanf( #arithmos korifon)

g->array =(int\*\*)malloc(sizeof(int\*)\*g->N); # dinamiki desmeusi gia korifes

for(i=1 ; i<g->N; i++){

g->array[i]= (int\*)malloc(sizeof(int)\*g->N); #arxikopoiisi olon ton stilwn(pithanes xwres

for(j=0;j<g->N;j++){ #arxikopoiisi olon ton diadromwn me 0

g->array[i][j]=0;

}

}

while(fscanf(fo,"%d %d %d", &i , &j , &w) !=EOF ){

add\_akmi(\*g , i,j, w);

printf(#oti diavase apo to arxio ) ; //gia skopous epiveveosis kai elegxou

}

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

ektipwsi\_graphou(g){

printf("%3d", g.array[i][j]);

}

