DMS 2nd year project

To check whether given graph is a POSET

//Coded in Code Blocks software

//DMS 2nd year project

#include<stdio.h>

#include<stdlib.h>

struct Node{

int info;

struct Node\* llink;

struct Node\* rlink;

};

typedef struct Node\* node;

int check=0;

node insert(node H){

int s;

printf("Enter item: ");

scanf("%d",&s);

node NN = (node)malloc(sizeof(struct Node));

NN->info = s;

NN->llink=NN->rlink=NULL;

if(H==NULL)

return NN;

node L=H;

while(L->llink!=NULL)

L=L->llink;

L->llink=NN;

return H;

}

void link(node H){

int m,n;

node L=H;

printf("Enter directed line: ");

scanf("%d%d",&m,&n);

while(L->info!=m)

L=L->llink;

node M=L;

while(M->rlink!=NULL)

M=M->rlink;

node NN = (node)malloc(sizeof(struct Node));

NN->info=n;

NN->llink=NN->rlink=NULL;

M->rlink=NN;

NN->llink=M;

}

void display(node H){

// int s,m;

// node L=H;

// printf("Enter the number whose directed lines are to be displayed: ");

// scanf("%d",&s);

// while(L->info!=s)

// L=L->llink;

// node M=L->rlink;

// while(M!=NULL){

// printf("(%d, %d)\n",s,M->info);

// M=M->rlink;

// }

node L=H, M;

printf("The realtion from the graph is\n R = { ");

while(L!=NULL){

M=L->rlink;

while(M!=NULL){

printf("(%d, %d) ",L->info, M->info);

M=M->rlink;

}

L=L->llink;

}

printf(" }\n");

}

void reflexive(node H){

node L=H, M;

while(L!=NULL){

M=L->rlink;

check=0;

while(M!=NULL){

if(L->info == M->info){

printf("(%d, %d) belongs to relation R\n",L->info, M->info);

check=1;

break;

}

M=M->rlink;

}

if(check==0){

printf("(%d, %d) does not belong to relation R\n",L->info, L->info);

return;

}

L=L->llink;

}

}

void antisymmetric(node H){

node L=H, M;

check=1;

while(L!=NULL){

M=L->rlink;

while(M!=NULL){

if(L->info != M->info){

node P=H, Q;

while(P!=NULL){

if(P->info == M->info) break;

P=P->llink;

}

Q=P->rlink;

while(Q!=NULL){

if(Q->info == L->info){

printf("(%d, %d) belongs to relation R but (%d, %d) also belongs to relation R\n",L->info, M->info, M->info, L->info);

check=0;

break;

}

Q=Q->rlink;

}

if(check==0) return;

printf("(%d, %d) belongs to relation R and (%d, %d) doesnot belong to relation R\n",L->info, M->info,M->info, L->info);

}

M=M->rlink;

}

L=L->llink;

}

}

void transitive(node H){

node A, B, C, D, E, F;

A=H;

check=1;

int i=0;

while(A!=NULL){

B=A->rlink;

E=A->rlink;

while(B!=NULL){

C=H;

if(B->info !=A->info){

while(C!=NULL){

if(C->info==B->info)

break;

C=C->llink;

}//C

D=C->rlink;

while(D!=NULL){

if(C->info != D->info){

check=0;

E=A->rlink;

while(E!=NULL){

if(E->info==D->info){

check=1;

break;

}

E=E->rlink;

}//E

// E=A->rlink;

if(check==0) {

printf("(%d, %d) and (%d, %d) belongs to relation R but (%d, %d) doesnot belong to relation R\n",A->info, B->info, C->info, D->info, A->info, D->info);

return;

}

printf("(%d, %d) and (%d, %d) belongs to relation R and (%d, %d) also belongs to relation R\n",A->info, B->info, C->info, D->info, A->info, D->info);

}//if

D=D->rlink;

}//D

}//if

B=B->rlink;

}//B

A=A->llink;

}//A

}

void main(){

int a,b;

node head=NULL;

for(;;){

printf("Enter nodes or links or check whether the graph is POSET: ");

scanf("%d",&a);

switch(a){

case 1: head = insert(head); break;

case 2: link(head); break;

case 3: display(head);

reflexive(head);

if(check==0){

printf("Hence it is not reflexive....and the Graph is not a POSET\n");

exit(0);

}

else

printf("Hence, The Graph is reflexive\n");

antisymmetric(head);

if(check==0){

printf("Hence it is not antisymmetric....and the Graph is not a POSET\n");

exit(0);

}

else

printf("Hence, The Graph is antisymmetric\n");

transitive(head);

if(check==0){

printf("Hence it is not transitive....and the Graph is not a POSET\n");

exit(0);

}

else

printf("Hence, The Graph is transitive\n");

printf("The graph is a POSET\n");

exit(0);

case 4: display(head); break;

default: exit(0);

}

}

}