GRAFURI

```
#include <stdio.h>
#include<iostream>
using namespace std;

struct nodStiva
{
	int inf;
	nodStiva *next;
};

struct nodCoada
{
	int inf;
	nodCoada *next;
};

void push(nodStiva **varf, int val)
{
	nodStiva *nou = (nodStiva*)malloc(1 * sizeof(nodStiva));
	nou->inf = val;
	nou->next = NULL;
	if (*varf == NULL) //daca e nula il creez
		*varf = nou;
	else
	{
		nou->next = *varf;   //daca nu, il pun inaintea lui
		*varf = nou;
	}
}
void pop(nodStiva **varf, int *val)
{
	if (*varf == NULL)
		return;                     //daca nu exista, nui am ce returna
	else
	{
		*val = (*varf)->inf;
		nodStiva *aux = *varf;
		*varf = (*varf)->next;
		free(aux);
	}
}

void put(nodCoada **prim, nodCoada **ultim, int val)//adauga in coada
{
	nodCoada *nou = (nodCoada*)malloc(1 * sizeof(nodCoada));
	nou->inf = val;
	nou->next = NULL;
	if (*prim == NULL && *ultim == NULL)
	{
		*prim = nou;
		*ultim = nou;
	}
	else
	{
```

```
		(*ultim)->next = nou;
		*ultim = nou;
	}
}

void get(nodCoada **prim, nodCoada **ultim, int* val)//extrage din coada
{
	if (*prim != NULL && *ultim != NULL)
	{
		*val = (*prim)->inf;
		nodCoada *aux = *prim;
		*prim = (*prim)->next;
		free(aux);
	}
	if (*prim == NULL)
		*ultim = NULL;
}
int main(void)
{
	FILE *f = fopen("fisier.txt", "r");
	int nr;
	fscanf(f, "%d", &nr); //numar noduri

	int  **mat = (int **)malloc(nr * sizeof(int *));
	for (int i = 0; i < nr; i++)
		mat[i] = (int *)malloc(nr * sizeof(int));

	for (int i = 0; i < nr; i++)
		for (int j = 0; j < nr; j++)
			mat[i][j] = 0;

	int n; //numar arce
	int ii; //nod sursa
	int jj; //nod desitnatie
	int g; //greutate


	fscanf(f, "%d", &n);//nr arce
	for (int i = 0; i < n; i++)
	{
		fscanf(f, "%d", &ii);
		fscanf(f, "%d", &jj);
		fscanf(f, "%d", &g);
		if (ii <= n && jj <= n)
		{
			mat[ii - 1][jj - 1] = g; //-1 pentru ca in vector incep de la 0
			mat[jj - 1][ii - 1] = g;
		}
	}
	fclose(f);

	//Afisare matrice
	for (int i = 0; i < nr; i++)
	{
		for (int j = 0; j < nr; j++)
		{
			printf("%d ", mat[i][j]);
```

```
	}
	printf("\n");
}

//vector noduri vizitate
int *vizitat = (int*)malloc(nr * sizeof(int));
for (int i = 0; i < nr; i++)
	vizitat[i] = 0;


//###########Stiva###########
nodStiva *varf = NULL;

int nod;
printf("\nParcurgere in adancime de la nodul: ");
scanf("%d", &nod);

push(&varf, nod);
vizitat[nod] = 1;

//imi parcurge in adancime stiva. imi scoate fiecare nod din stiva
while (varf != NULL)
{
	pop(&varf, &nod);
	printf("%d-", nod+1);
	for(int k = 0; k < nr; k++)
		if (mat[nod][k] !=0 && vizitat[k] == 0)
		{
			push(&varf, k);
			vizitat[k] = 1;
		}
}

//##############Coada############


for (int i = 0; i < nr; i++)
	vizitat[i] = 0;

printf("\nParcurgere in latime de la nodul: ");
scanf("%d", &nod);
nodCoada* prim = NULL, *ultim = NULL;
put(&prim, &ultim, nod);
vizitat[nod] = 1;

while (prim != NULL)
{
	get(&prim,&ultim, &nod);
	printf("%d-", nod + 1);
	for (int k = 0; k < nr; k++)
		if (mat[nod][k]  != 0 && vizitat[k] == 0)
		{
			put(&prim,&ultim, k);
			vizitat[k] = 1;
		}
}
```

```
        return 0;
}
```