#include<stdio.h>

#include<stdlib.h>

#include<math.h>

int min(int a, int b){

return (a<b)?a:b;

}

int max1(int a, int b){

return (a>b)?a:b;

}

int networkDelayTime(int\*\* times, int timesSize, int\* timesColSize, int n, int k)

{

int max=10000;

int i,j,u,v,w;

int \*dist=(int \*)malloc(n\*sizeof(int));

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

{

dist[i]=max;

}

dist[k-1]=0;

/\*

for(int f=0; f<n; f++){

printf("%d ", dist[f]);

}

printf("\n");

\*/

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

{

for(j=0;j<timesSize;j++)

{

u=times[j][0]-1;

v=times[j][1]-1;

w=times[j][2];

dist[v]=min(dist[v],dist[u]+w);

}

/\*

for(int f=0; f<n; f++){

printf("%d ", dist[f]);

}

printf("\n");

\*/

}

int ans=0;

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

{

if(dist[i]==max)

{

return -1;

}

ans=max1(ans,dist[i]);

}

return ans;

}

int main()

{

int n,k,ts,tcs,i,j;

printf("enter no of nodes(n) : ");

scanf("%d",&n);

printf("enter starting node(k) : ");

scanf("%d",&k);

printf("enter no of edges : ");

scanf("%d",&ts);

printf("enter no of elements in each edge[] : ");

scanf("%d",&tcs);

printf("enter elements : \n");

int\*\* arr = (int\*\*)malloc(ts \* sizeof(int\*));

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

arr[i] = (int\*)malloc(tcs \* sizeof(int));

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

printf("Enter edge %d \n", i+1);

scanf("%d %d %d",&arr[i][0], &arr[i][1], &arr[i][2]);

}

int ans=networkDelayTime(arr,ts,&tcs,n,k);

printf("Minimum time is : %d \n",ans);

}

//[[2,1,1],[2,3,1],[3,4,1]]

//4

//2

//3

//3

Output:

enter no of nodes(n) : 4

enter starting node(k) : 2

enter no of edges : 3

enter no of elements in each edge[] : 3

enter elements :

Enter edge 1

2 1 1

Enter edge 2

2 3 1

Enter edge 3

3 4 1

Minimum time is : 2