Max and min

Code:

#max.c

#include<stdio.h>

#include <limits.h>

int main(){

int nT,nM;//number of tasks , number of machines

printf("\nEnter number of machines and tasks\n");

scanf("%d%d",&nM,&nT);

/\*

Declare a 2d-array of size nM x nT

Data should be in the following format :

T1 T2 T3

M1 | 140 | 20 | 60 |

M2 | 100 | 100 | 70 |

\*/

int minMax[nM][nT];

int tmp[nM][nT];

int makespan=0;

printf("\nFill Data\n");

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

for(int j=0;j<nT;j++){

scanf("%d",&minMax[i][j]);

tmp[i][j]=minMax[i][j];

}

// visualise data

printf("\nOriginal Data\n");

for(int i=0;i<nM;i++){

for(int j=0;j<nT;j++)

printf("%d ",minMax[i][j]);

printf("\n");

}

//This array will hold the answer

int resultTask[nT];

int resultMachine[nT];

int resultTime[nT];

int ptr=-1; //Indicates if result set is full or not

while(ptr<nT-1){

int time[nT],machine[nT]; //stores minimum time w.r.t machine of each task

for(int j=0;j<nT;j++){

int maximum = INT\_MIN;

int pos=-1;

for(int i=0;i<nM;i++){

if(minMax[i][j]>maximum){

maximum=minMax[i][j];

pos=i;

}

}

time[j]=maximum;

machine[j]=pos;

}

// Now we find task with minimum time

int minimum=INT\_MAX;

int pos=-1;

for(int j=0;j<nT;j++){

if(time[j]<minimum && time[j] != INT\_MIN){

minimum=time[j];

pos=j;

}

}

resultTask[++ptr]=pos;

resultMachine[ptr]=machine[pos];

resultTime[ptr]=tmp[machine[pos]][pos];

if(minimum>makespan)

makespan=minimum;

// resetting states

for(int i=0;i<nM;i++){

for(int j=0;j<nT;j++){

if(j==resultTask[ptr])

minMax[i][j]=INT\_MIN;

else if(i==resultMachine[ptr] && minMax[i][j]!=INT\_MIN)

minMax[i][j]+=minimum;

else

continue;

}

}

}

//printing answer

printf("\nScheduled Task are :\n");

for(int i=0;i<nT;i++){

printf("\nTask %d Runs on Machine %d with Time %d units\n",resultTask[i]+1,resultMachine[i]+1,resultTime[i]);

}

printf("\nMakeSpan time : %d units\n",makespan);

return 0;

}

Min.c

#include<stdio.h>

#include <limits.h>

int main(){

int nT,nM;//number of tasks , number of machines

printf("\nEnter number of machines and tasks\n");

scanf("%d%d",&nM,&nT);

/\*

Declare a 2d-array of size nM x nT

Data should be in the following format :

T1 T2 T3

M1 | 140 | 20 | 60 |

M2 | 100 | 100 | 70 |

\*/

int minMin[nM][nT];

int tmp[nM][nT];

int makespan=0;

printf("\nFill Data\n");

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

for(int j=0;j<nT;j++){

scanf("%d",&minMin[i][j]);

tmp[i][j]=minMin[i][j];

}

// visualise data

printf("\nOriginal Data\n");

for(int i=0;i<nM;i++){

for(int j=0;j<nT;j++)

printf("%d ",minMin[i][j]);

printf("\n");

}

//This array will hold the answer

int resultTask[nT];

int resultMachine[nT];

int resultTime[nT];

int ptr=-1; //Indicates if result set is full or not

while(ptr<nT-1){

int time[nT],machine[nT]; //stores minimum time w.r.t machine of each task

for(int j=0;j<nT;j++){

int minimum = INT\_MAX;

int pos=-1;

for(int i=0;i<nM;i++){

if(minMin[i][j]<minimum){

minimum=minMin[i][j];

pos=i;

}

}

time[j]=minimum;

machine[j]=pos;

}

// Now we find task with minimum time

int minimum=INT\_MAX;

int pos=-1;

for(int j=0;j<nT;j++){

if(time[j]<minimum){

minimum=time[j];

pos=j;

}

}

resultTask[++ptr]=pos;

resultMachine[ptr]=machine[pos];

resultTime[ptr]=tmp[machine[pos]][pos];

if(minimum>makespan)

makespan=minimum;

// resetting states

for(int i=0;i<nM;i++){

for(int j=0;j<nT;j++){

if(j==resultTask[ptr])

minMin[i][j]=INT\_MAX;

else if(i==resultMachine[ptr] && minMin[i][j]!=INT\_MAX)

minMin[i][j]+=minimum;

else

continue;

}

}

}

//printing answer

printf("\nScheduled Task are :\n");

for(int i=0;i<nT;i++){

printf("\nTask %d Runs on Machine %d with Time %d units\n",resultTask[i]+1,resultMachine[i]+1,resultTime[i]);

}

printf("\nMakespan : %d units\n",makespan);

return 0;

}