// **bidirectional bfs**

queue< pair<int,int> >q,q2;

map< pair<int,int>,pair<int,int> > parent,parent2;

int vis[100][100],vis2[100][100];

void revpath(){

int x1=destmid.first;int y1=destmid.second;

while(x1!=dest.first or y1!=dest.second){

pair<int,int> temp=parent2[{x1,y1}];

cout<<x1<<" "<<y1<<" "<<temp.first<<" "<<temp.second<<endl;

parent[temp]={x1,y1};

x1=temp.first,y1=temp.second;

}

}

void bfs(){

if(q.size()!=0 and found==false){

pair<int,int> u= q.front();

int x=u.first; int y=u.second;

q.pop();

for(int i=0;i<4;i++){

int ax=x+dx[i]; int ay=y+dy[i];

if(ax>=0 and ay>=0 and ax<=80 and ay<=37 and vis[ax][ay]==0 and vis2[ax][ay]==0){

vis[ax][ay]=1000;

// if(dest==make\_pair(ax,ay) ) found=true;

parent[{ax,ay}]={x,y};

q.push({ax,ay});

}

else if(ax>=0 and ay>=0 and ax<=80 and ay<=37 and vis2[ax][ay]!=0 and vis[ax][ay]==0) destmid={ax,ay},parent[{ax,ay}]={x,y},found=true,revpath();

}

}

}

void bfs2(){

if(q2.size()!=0 and found==false){

pair<int,int> u= q2.front();

int x=u.first; int y=u.second;

q2.pop();

for(int i=0;i<4;i++){

int ax=x+dx[i]; int ay=y+dy[i];

if(ax>=0 and ay>=0 and ax<=80 and ay<=37 and vis2[ax][ay]==0 and vis[ax][ay]==0){

vis2[ax][ay]=1000;

//if(dest==make\_pair(ax,ay) ) found=true;

parent2[{ax,ay}]={x,y};

q2.push({ax,ay});

}

//else if(ax>=0 and ay>=0 and ax<=80 and ay<=37 and vis2[ax][ay]==0 and vis[ax][ay]!=0) destmid={ax,ay},found=true;

}

}

}