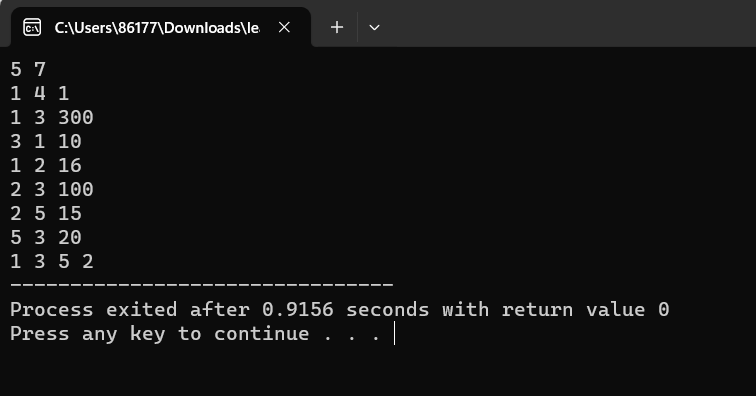
Task1



Task2

#include <iostream>

#include<vector>

using namespace std;

int inf = 99999999;

int f[20][20];

int dist[20];

int n ;

int parent[20];

int total=0;

class Solution {

public:

int getMin(){

    int k=0;

    for(int i=1;i<=n;i++){

        if(dist[i]<dist[k]&&dist[i]>0)  k=i;

    }

    return k;

}

void Prim(int i){

    while(true){

        int v=getMin();

        //cout<<"v:"<<v<<endl;

        if(v==0) break;

        if(v!=i) total+=dist[v];

        //cout<<v<<" "<<total<<endl;

        dist[v]=0;

        for(int i=1;i<=n;i++){

            if(f[i][v]!=inf){

                if(dist[i]!=0){

                    if(f[i][v]<dist[i]){dist[i]=f[i][v];parent[i]=v;}

                }

            }

        }

        for(int i=1;i<=n;i++) cout<<dist[i]<<" ";

        cout<<endl;

    }

}

    int minCostConnectPoints(vector<vector<int>>& points) {

        n=points.size();

        for(int i=1;i<=n;i++){

            dist[i]=inf;

            for(int j=i;j<=n;j++){

                f[i][j]=f[j][i]=abs(points[i-1][1]-points[j-1][1])+abs(points[i-1][0]-points[j-1][0]);

            }

        }

       /\* for(int i=1;i<=n;i++){

            for(int j=1;j<=n;j++){

                cout<<f[i][j]<<"\t";

            }

            cout<<endl;

        }\*/

        for(int i=2;i<=n;i++){

        if(f[1][i]<dist[1]) dist[1]=f[1][i];

    }

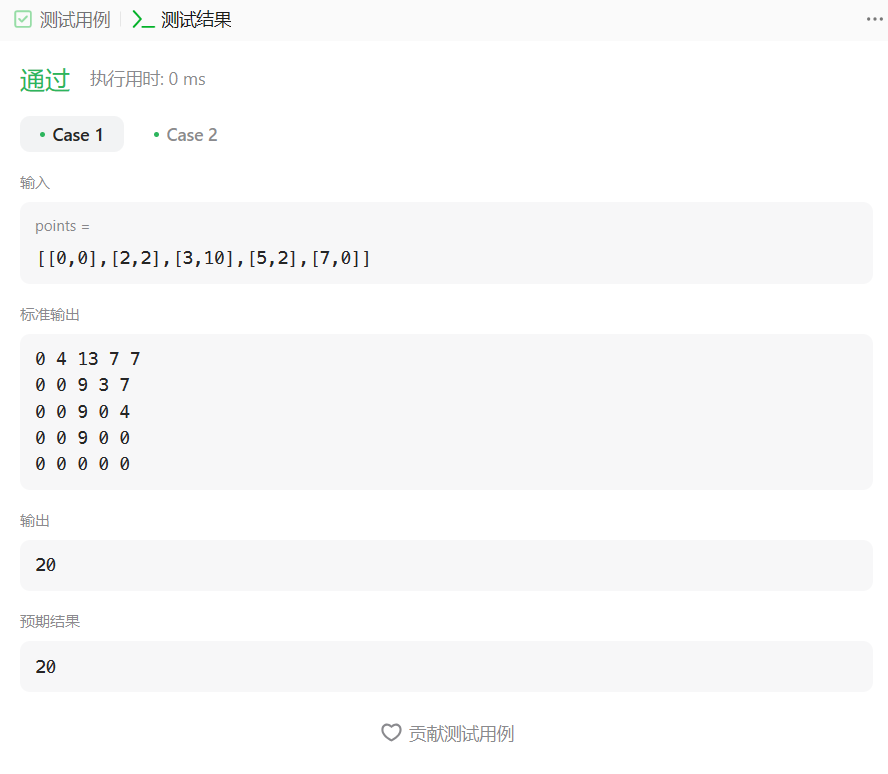
    dist[0]=inf;

    Prim(1);

        return total;

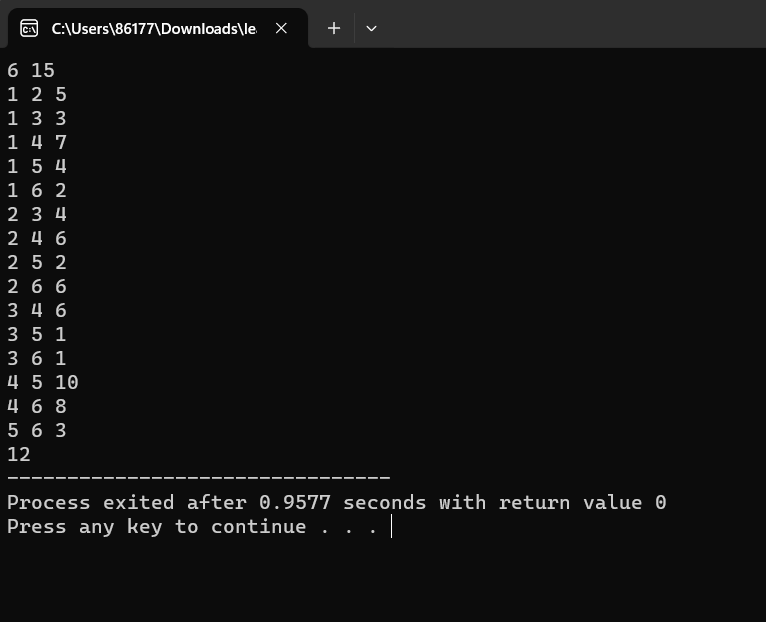
    }

};





Task3



Task4

