#include <stdio.h>

#include <math.h>

#define INF 1000

#define DEFAULT 0

struct position{

float x;

float y;

float theta;

int index;

}pos[10];

int posNum = 0;

int stack[10] = { 0, };

int top = 0;

FILE \*inF, \*outF;

float Distance(int index1, int index2);

void swap(int indexA, int indexB);

int pop();

void push(int val);

void printStack();

void main()

{

//pos[0].x = 0;

//pos[0].y = 0;/\*start at (0,0)\*/

inF = fopen("input.txt", "r");

fscanf(inF, "%d", &posNum);

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

{

fscanf(inF, "%f %f", &pos[i].x, &pos[i].y);

printf("\n%2.1f %2.1f", pos[i].x, pos[i].y);

pos[i].index = i;

pos[i].theta = atan2(pos[i].y, pos[i].x);

}

fclose(inF);

for (int i = posNum-1; i >0; i--)//bubble sorting

{

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

{

if (pos[j].theta>pos[j + 1].theta)

{

swap(j, j + 1);

}

}

}

printf("\nSorted: ");

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

{

printf("\n%d(%2.1f,%2.1f) ,theta(y,x) is %f", i, pos[i].x, pos[i].y, pos[i].theta);

}

float distance = 0;

for (int i = 1; i <= 10; i++)//initialize stack

{

stack[i] = DEFAULT;

}

int stackedNum = 0;

while (pos[top].index!=pos[posNum].index)

{

push(++stackedNum);

if (top <= 2)

{

printf("\n%d,%d", top, stackedNum);

printStack();

continue;

}

while (getTheta(top-1,top)<=getTheta(top-2,top-1))

{

printStack();

printf("-->pos[%d] is popped!", top -1);

int first = stack[top];

pop();

int second = stack[top];

pop();

push(first);

}

printStack();

}

int cur = 0;

while (cur < top)//calculate distance

{

printf("\ndistance(%d,%d):%4.2f", pos[cur].index,pos[cur+1].index,Distance(stack[cur], stack[cur + 1]));

distance += Distance(stack[cur], stack[cur + 1]);

cur++;

}

outF = fopen("output.txt", "w");

fprintf(outF, "distance: %4.2f", distance);

fclose(outF);

}

float getTheta(int indexA, int indexB)

{

return (atan2(pos[indexB].y - pos[indexA].y, pos[indexB].x - pos[indexA].x));

}

float Distance(int index1, int index2)

{

float difX = pos[index2].x - pos[index1].x;

float difY = (pos[index2].y - pos[index1].y);

return sqrt(pow(difX, 2) + pow(difY, 2));

}

void swap(int indexA, int indexB)

{

struct position temp;

temp.x = pos[indexA].x;

pos[indexA].x = pos[indexB].x;

pos[indexB].x = temp.x;

temp.y = pos[indexA].y;

pos[indexA].y = pos[indexB].y;

pos[indexB].y = temp.y;

temp.theta = pos[indexA].theta;

pos[indexA].theta = pos[indexB].theta;

pos[indexB].theta = temp.theta;

temp.index = pos[indexA].index;

pos[indexA].index = pos[indexB].index;

pos[indexB].index = temp.index;

}

void push(int val)

{

stack[++top] = val;

}

int pop()

{

int temp = stack[top];

stack[top--] = DEFAULT;

return temp;

}

void printStack()

{

printf("\nStack[");

for (int k = 0; k <= top; k++)

{

printf("%2.1f, ", pos[stack[k]].x);

}

printf("]");

}