

Lab SSC0220 – 05/09/2014**Conhecimentos básicos para desenvolver as tarefas de hoje**

Distância euclidiana entre dois pontos: A distância d entre dois pontos $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$ é dada pela seguinte fórmula:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

- Um exemplo:

Dados os pontos $P_1 = (0, 5)$ e $P_2 = (13, 16)$, a distância entre eles é 17,02

Estruturas de dados a serem utilizadas nas tarefas seguintes:

```
typedef struct pnt {  
    int x, y;  
} ponto;
```

```
typedef struct vrtc{  
    ponto p;  
    struct vrtc* prox;  
} vertice;
```

```
typedef struct fig{  
    int N;  
    int Dy, Dx;  
    vertice* p_inicio;  
} figura;
```

Tarefa 1

Buscando um ponto

Tarefa 1a) Busca um ponto: Crie o corpo da função `busca` descrita a seguir:

```
vertice* busca (int pos, figura *f)
```

A função deverá receber um inteiro indicando a posição que um vértice ocupa na lista do tipo `vertice`, existente em uma estrutura do tipo `figura`. A função deve retornar um ponteiro do tipo `vertice` que apontará para o elemento encontrado. Quando não existir o elemento, retorne `NULL`. Considere que a primeira posição é a posição de número 1.

Tarefa 1b) Programa principal: Chegou a hora de criar o programa principal que chamará a função `busca` criada anteriormente. O programa deverá ler um valor inteiro N e os pontos referentes aos N vértices da figura geométrica a ser dada. Sua tarefa é, dado P posições quaisquer, indicar os pontos que estão nelas.

Entrada

O arquivo de entrada contém diversos casos de teste. A primeira linha de cada caso de teste contém um inteiro N indicando a quantidade de vértices ($0 \leq N \leq 10$) da figura. A próxima linha contém $N*2$ inteiros, x_i e y_i , separados por espaços únicos, indicando os pontos dos vértices ($-10^4 \leq x_i \leq 10^4$, $-10^4 \leq y_i \leq 10^4$). A linha seguinte contém um valor inteiro, indicando a quantidade P de posições a serem procuradas. As B linhas seguintes ($0 \leq B \leq 100$) contém as posições dos vértices a serem procurados. O fim da entrada é indicado por uma linha contendo o valor 0, indicando que $N=0$.

Saída

Para cada caso de teste da entrada, imprima B linhas, indicando o valor de x e y dos vértices nas posições fornecidas, separados por um espaço único. Quando a posição fornecida não existir imprima a frase "Nao achou".

Exemplo de entrada	Saída para o exemplo de entrada
--------------------	---------------------------------

```
3
-3 2 -2 5 -5 3
3
1
2
3
5
8 6 5 5 3 3 4 0 7 2
3
3
2
1
2
8 6 5 5
1
4
0
```

```
-3 2
-2 5
-5 3
3 3
5 5
8 6
Nao achou
```

Tarefa 2

Distância entre pontos

Crie o corpo da função `distancia` descrita a seguir:

```
float distancia (ponto p, ponto p)
```

A função deverá receber duas estruturas do tipo `ponto` e retornar a distância entre os pontos dados.

Programa principal: Chegou a hora de criar o programa principal que chamará a função `distancia`. O programa deverá ler um valor inteiro N e os pontos referentes aos N vértices da figura geométrica a ser dada. Sua tarefa é, dadas posições de dois vértices, indicar qual a distância entre eles.

Entrada

O arquivo de entrada contém diversos casos de teste. A primeira linha de cada caso de teste contém um inteiro N indicando a quantidade de vértices ($0 \leq N \leq 10$) da figura. A próxima linha contém $N*2$ inteiros, x_i e y_i , separados por espaços únicos, indicando os pontos dos vértices ($-10^4 \leq x_i \leq 10^4$, $-10^4 \leq y_i \leq 10^4$). A linha seguinte contém a posição P do vértice ($1 \leq P \leq N$) que será utilizado para verificar as distâncias com todos os vértices. O fim da entrada é indicado por uma linha contendo o valor 0.

Saída

Para cada caso de teste da entrada, imprima 1 linha, indicando a distância entre os vértices pertencentes às posições fornecida. A distância deve ser impressa considerando somente duas casas decimais. Quando uma ou mais posições fornecidas não existir imprima a frase "Nao achou".

Exemplo de entrada	Saída para o exemplo de entrada
--------------------	---------------------------------

```
3
-3 2 -2 5 -5 3
2 3
5
8 6 5 5 3 3 4 0 7 2
5 1
4
0 0 5 6 8 6 5 5
1 3
3
4 6 7 2 1 9
1 4
0
```

```
3.61
4.12
10.00
Nao achou
```

Tarefa 3

Distância entre um vértice e os demais

Crie o corpo da função `distancia_todos` descrita a seguir:

```
float* distancia_todos (int pos, figura *f)
```

A função deverá receber uma estrutura do tipo `figura` que conterá uma lista com todos os vértices da figura e um inteiro indicando a posição de um elemento da lista. A função deve retornar um vetor contendo as distâncias do vértice na posição fornecida para todos os vértices da figura, quando a posição encontrada existir. Quando não existir o elemento, retorne NULL. Considere que a primeira posição é a posição de número 1. Utilize a função `distancia` criada anteriormente.

Programa principal: Chegou a hora de criar o programa principal que chamará a função `distancia_todos` criada anteriormente. O programa deverá ler um valor inteiro N e os pontos referentes aos N vértices da figura geométrica a ser dada. Sua tarefa é, dado um dos vértices, indicar qual a sua distância para os N vértices pertencentes a uma figura.

Entrada

O arquivo de entrada contém diversos casos de teste. A primeira linha de cada caso de teste contém um inteiro N indicando a quantidade de vértices ($0 \leq N \leq 10$) da figura. A próxima linha contém $N*2$ inteiros, x_i e y_i , separados por espaços únicos, indicando os pontos dos vértices ($-10^4 \leq x_i \leq 10^4$, $-10^4 \leq y_i \leq 10^4$). A linha seguinte contém a posição P do vértice ($1 \leq P \leq N$) que será utilizado para verificar as distâncias com todos os vértices. O fim da entrada é indicado por uma linha contendo o valor 0.

Saída

Para cada caso de teste da entrada, imprima 1 linha, indicando a distância do vértice para todos os vértices da figura. A distância deve ser impressa considerando somente duas casas decimais. Quando a posição fornecida não existir imprima a frase "Nao achou".

Exemplo de entrada	Saída para o exemplo de entrada
--------------------	---------------------------------

```
3
-3 2 -2 5 -5 3
2
5
8 6 5 5 3 3 4 0 7 2
5
4
0 0 5 6 8 6 5 5
1
3
4 6 7 2 1 9
4
0
```

```
3.16 0.00 3.61
4.12 3.61 4.12 3.61 0.00
0.00 7.81 10.00 7.07
Nao achou
```