

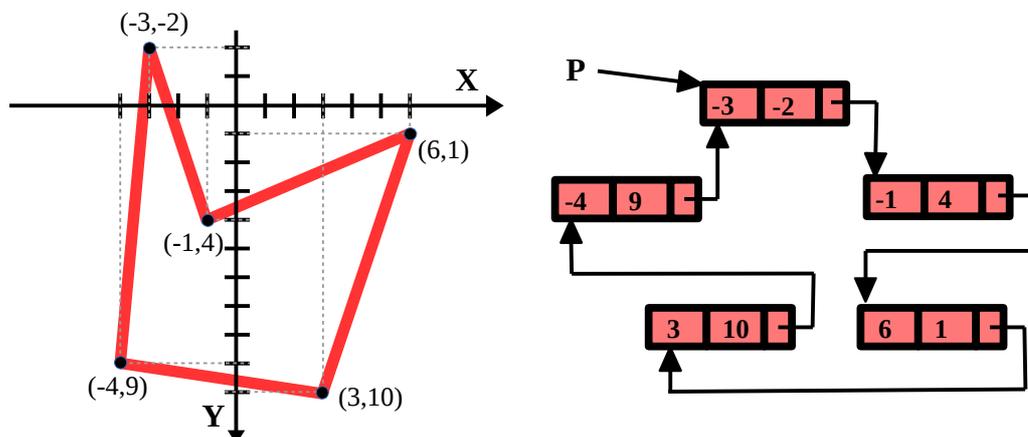
MAC122 Princípios de Desenvolvimento de Algoritmos

EP 01

Prof.: Paulo Miranda
Instituto de Matemática e Estatística (IME)
Universidade de São Paulo (USP)

Estruturas ligadas:

Um polígono pode ser representado pela sequência de seus vértices, armazenados em uma lista circular em sentido horário. Exemplo:



Considere a seguinte estrutura:

```
struct Vertice{  
    double x;  
    double y;  
    struct Vertice *prox;  
};  
typedef struct Vertice* Poligono;
```

Faça um programa que lê da entrada padrão um polígono fornecido, na forma de um inteiro $n > 2$ (número de lados) seguido das n coordenadas em sentido horário de seus vértices (pares x e y do tipo `double`), e que calcula e imprime seu perímetro e área com uma casa decimal.

OBS: O polígono deve ser armazenado em uma lista circular alocada dinamicamente. O apontador para a lista pode apontar para qualquer um dos vértices do polígono gerado (não existe uma ordem preferencial). Toda memória da lista deve ser liberada ao final do programa.

Você deve implementar as seguintes funções auxiliares:

```
int NumeroDeVertices(Poligono P);
```

Função que conta o número de vértices de um polígono fornecido. Por exemplo, no caso da lista circular do exemplo acima, a função deve devolver 5.

```
double Distancia(struct Vertice *A, struct Vertice *B);
```

Função que devolve a distância euclidiana entre dois vértices fornecidos.

```
double Perimetro(Poligono P);
```

Função que calcula e devolve o perímetro do polígono fornecido. Por exemplo, no caso da lista circular do exemplo acima, a função deve devolver 41,54359.

```
double Area(Poligono P);
```

Função que calcula e devolve a área do polígono fornecido. Por exemplo, no caso da lista circular do exemplo acima, a função deve devolver 60,0.

Dica: Consulte o capítulo 2.6 do livro "Introdução à Geometria Computacional" de Luiz Henrique de Figueiredo e Paulo César Pinto Carvalho (ISBN 85-244-0061-7).

```
void LiberaPoligono(Poligono P);
```

Função que libera a memória alocada do polígono fornecido.

A seguir são apresentados alguns exemplos de execução do programa. As entradas do usuário correspondem aos textos em vermelho e as saídas do programa aos textos em azul.

Exemplo 1:

```
5
-3 -2
-1 4
6 1
3 10
-4 9
Perimetro: 41.5
Area: 60.0
```

Exemplo 2:

```
4
50.0 50.0
600.0 50.0
600.0 400.0
50.0 400.0
Perimetro: 1800.0
Area: 192500.0
```