Faculdade de Computação

Programação Procedimental

8° Laboratório de Programação C

Prof. Cláudio C. Rodrigues

1. Introdução

As estruturas (*structs*) permitem o agrupamento de variáveis de vários tipos sob o mesmo nome. E esse mesmo nome passará ser um novo tipo de dados, tal como o int ou float.
Assim, podemos associar valores que tenham alguma relação lógica, por exemplo, podemos caracterizar uma entidade pessoa com as seguintes propriedades: um atributo int para idade e um atributo string para o nome.

Sintaxe:

```
struct <identificador> {
  <tipo>  campo_um   ;
  <tipo>  campo_dois ;
};
```

---

2. Problemas

P1. Escreva a função simplifica que receba dois inteiros **a** e **b** e devolve como resposta um número racional (*p/q*) que representa a fração **a/b**. Simplificar uma fração é encontrar outra fração equivalente dividindo o numerador e o denominador da fração por um mesmo número não-nulo. O máximo divisor comum de **a** e **b** é o elemento divisor que garante ao campo **q** de todo racional seja estritamente positivo e que o mdc de **p** e **q** é **1**.

Programa 2:

```
#include <stdlib.h>
#include <stdio.h>
typedef struct {
       int p;
       int q;
}Racional;
Racional simplifica(int a, int b);
int mdc(int a, int q);
```

```
int main(int argc, char *argv[]){
     Racional r;
     r = simplifica(4,12);
     Printf("%d/%d\n",r.p,r.q);
     return 0;
}
```

O algoritmo deve estar contido no arquivo "**racional.c**".

P2. No plano cartesiano, um ponto **P** é caracterizado pelo par de coordenadas (x, y):

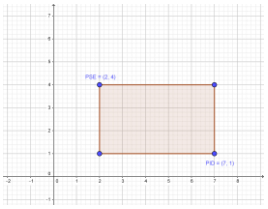
a) Defina uma estrutura chamada ***Ponto*** para armazenar as coordenadas de um ponto do plano cartesiano.

b) Escreva uma função chamada ***distancia*** que receba dois argumentos **A** e **B** do tipo ***Ponto*** e retorna a distância euclidiana entre eles, segundo a fórmula:

$$distancia(A,B) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

O algoritmo deve estar contido no arquivo "**ponto.c**".

P3. No sistema de coordenadas cartesianas, cada ponto é identificado por duas coordenadas reais, x e y, que definem a sua posição em relação à interseção de dois eixos perpendiculares (o eixo das abscissas – horizontal – e o eixo das ordenadas – vertical).



Codifique as estruturas e funções especificadas nos itens abaixo:

a) Defina em linguagem C um tipo estruturado ***Retangulo*** que represente um retângulo paralelo aos eixos do sistema de coordenadas cartesianas. Um retângulo pode ser representado por dois ***Pontos***: o ponto do canto superior esquerdo (pse) e o ponto do canto inferior direito (pid).

b) Declare e defina em linguagem C a função ***perímetro***() que recebe como parâmetro um Retangulo e retorna o perímetro do retângulo. Use o tipo Retangulo definido no problema.

c) Declare e defina em linguagem C a função ***area***() que recebe como parâmetro um Retangulo e retorna a área do retângulo. Use o tipo Retangulo definido no problema.

d) Declare e defina em linguagem C a função ***inRetangulo***() que recebe como parâmetros um Retangulo e um Ponto e retorna 1 se o Ponto pertence ao Retangulo, e 0 se não pertencer. Use os tipos Ponto e Retangulo já definidos no problema.

O algoritmo deve estar contido no arquivo "**retangulo.c**".

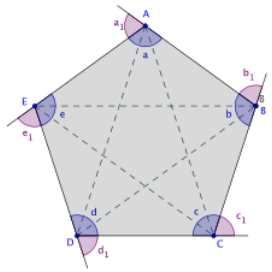
P4. Polígono é uma superfície plana limitada por segmentos de reta (arestas ou lados), cujos vértices são formados por duas arestas. Um polígono simples divide o plano em que se encontra em duas regiões (a interior e a exterior), isto é, bidimensional (eixo do "X" e do "Y"), sem pontos comuns.

Elementos de um polígono:

Um polígono estrelado é uma linha poligonal fechada não-simples com propriedades especiais.

Lados - cada um dos segmentos de reta que une vértices consecutivos:
$\overline{AB}, \overline{BC}, \overline{CD}, \overline{DE}, \overline{EA}$.

Vértices - Ponto de encontro dos segmentos: A, B, C, D, E.



- No sistema de coordenadas cartesianas, cada ponto é identificado por duas coordenadas reais, *x* e *y*, que definem a sua posição em relação à interseção de dois eixos perpendiculares (o eixo das abcissas – horizontal – e o eixo das ordenadas – vertical).
- Assim, um polígono pode ser representado como uma coleção de pontos que são seus vértices consecutivos.

Considere os códigos definidos no quadro abaixo. Codifique as seguintes funções:

a) Declare e defina o tipo estruturado ***Poligono***, como sendo uma coleção de vértices (Pontos) e a quantidade de vértices.

b) Declare e Defina em linguagem C a função *perimetro*() que recebe como parâmetro um polígono e retorna o valor do perímetro do polígono que é a soma da medida dos lados.

c) Declare e Defina em linguagem C a função *centroGeo*() que recebe como parâmetro um polígono e retorna um Ponto que é o centro geométrico do polígono.

- As coordenadas do centro geométrico podem ser determinadas pelas fórmulas:

$$x_{cg} = \frac{\sum_{i=1}^{n} x_i}{n} \quad y_{cg} = \frac{\sum_{i=1}^{n} y_i}{n}$$

Onde x_i e y_i são as coordenadas dos vértices do polígono e n a quantidade de vértices.

O algoritmo deve estar contido no arquivo "**poligono.c**".