

Algoritmos e Programação I

Profa. Luciana Montera

Faculdade de Computação - Facom/UFMS

Exercícios de Estrutura Condicional

Para as questões a seguir, apresente um algoritmo e uma implementação em python.

1. Leia dois números e imprima apenas o maior deles.
2. Leia um inteiro e determine se ele é par ou ímpar.
3. Leia um número e informe se ele é ou não múltiplo de cinco.
4. Leia um número e determine se ele é inteiro ou não. Considere que todo número que não tiver nenhum dígito após o ponto ou tiver apenas zeros é um inteiro. Exemplos:

Entrada: 7

Saída: É inteiro

Entrada: 31.80

Saída: Não é inteiro

Entrada: 4.76

Saída: Não é inteiro

Entrada: 7.00

Saída: É inteiro

Entrada: 12.00

Saída: É inteiro

Entrada: 12.01

Saída: Não é inteiro

5. Leia dois inteiros a e b e informe ao usuário se a é múltiplo de b .
6. Leia dois valores inteiros A e B . Se A e B forem iguais, some-os, caso contrário multiplique-os. Como saída você deve informar a operação realizada e o seu resultado.
7. Leia dois inteiros e um caracter “+”, “-”, “*” ou “/” representando as operações de soma, subtração e divisão, respectivamente. Em seguida realiza a operação desejada pelo usuário e o resultado conforme os exemplos a seguir:

Entrada: 8 2 *

Saída: (8) * (2) = 16

Entrada: 25 -7

Saída: (25) - (-7) = 18

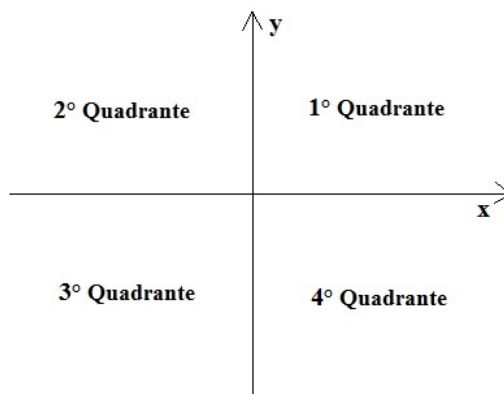
Os dados de entrada serão fornecidos todos em uma única linha.

8. A seguinte tabela serve de base para o cálculo do valor do imposto a pagar:

Base de Cálculo	Valor do Imposto
$\text{renda} < \text{R\$}1434.59$	isento
$\text{R\$ } 1434.60 \leq \text{renda} \leq \text{R\$}2150.00$	7.8
$\text{R\$ } 2150.01 \leq \text{renda} \leq \text{R\$}2866.70$	15.0
$\text{R\$ } 2866.71 \leq \text{renda} \leq \text{R\$}3582.00$	22.5
Acima de R\$ 3582.00	27.5

Leia a renda anual de um indivíduo e informe o valor do imposto a ser pago pelo mesmo.

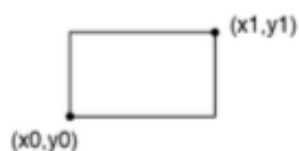
9. Leia duas coordenadas (x, y) que representam um ponto no plano cartesiano e determine se este ponto se encontra na origem, no eixo x, no eixo y ou em um dos quadrantes, conforme esquema da figura a seguir.



10. (OBI - fase 1 - 2007)¹ Detecção de colisão é uma das operações mais comuns (e importantes) em jogos eletrônicos. O objetivo, basicamente, é verificar se dois objetos quaisquer colidiram, ou seja, se a interseção entre eles é diferente de vazio. Isso pode ser usado para saber se duas naves colidiram, se um monstro bateu numa parede, se um personagem pegou um item, etc.

Para facilitar as coisas, muitas vezes os objetos são aproximados por figuras geométricas simples (esferas, paralelepípedos, triângulos etc). Neste problema, os objetos são aproximados por retângulos num plano 2D e seus lados do retângulo são sempre paralelos aos eixos x e y.

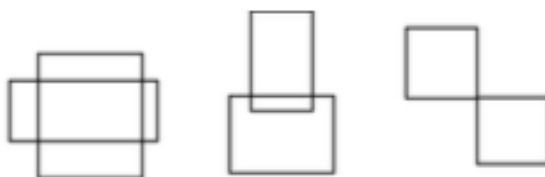
Escreva um programa que, dados dois retângulos, determine se eles se interceptam ou não. Cada retângulo é identificado pelo ponto do canto inferior esquerdo e pelo ponto do canto superior direito. Veja alguns esquemas na figura a seguir.



Duas coordenadas definem um retângulo



Exemplo de retângulos que não se interceptam



Exemplos de retângulos que se interceptam

¹com algumas adaptações

11. Faça um programa que receba a hora do início de um jogo e a hora do término (cada hora é composta por duas variáveis inteiras: hora e minuto). Calcule e mostre a duração do jogo (horas e minutos). Um jogo pode começar em um dia e terminar em outro. *Lembre-se:*
- (a) Se o segundo inicial for maior que o segundo final, então devemos somar 60 no segundo final e diminuir 1 do minuto final
 - (b) Se o minuto inicial for maior que o minuto final, então devemos somar 60 no minuto final e diminuir 1 da hora final
 - (c) Se a hora de início for maior que a hora de término, então é porque o jogo terminou no outro dia. Deve-se somar 24 horas à hora final

Segue alguns exemplos:

Entrada:

12 31 54

16 32 54

Saída: 4 1 0

Entrada:

20 30 25

7 12 39

Saída: 10 42 14

Entrada:

12 31 54

16 15 54

Saída: 3 44 0

Entrada:

16 54 33

8 37 21

Saída: 15 42 48