

## EXERCÍCIOS DE FORTRAN

### LISTA 01

1 – A função intrínseca **INT(x)** converte um número **x** real em inteiro, preservando apenas a parte inteira de **x** (truncando a parte decimal do **x**). A função **NINT(x)** converte o número **x** de real para inteiro arredondando-o para o valor inteiro mais próximo de **x**. Que valores serão exibidos pelo seguinte programa?

```
PROGRAM sample_1
  INTEGER :: i1, i2, i3
  REAL :: a1=2.4, a2
  i1 = a1
  i2 = INT(a1*i1)
  i3 = NINT(a1*i1)
  a2 = a1**i1
  WRITE(*,*) i1, i2, i3, a1, a2
END PROGRAM sample_1
```

2 – A função intrínseca **REAL(i)** converte o número inteiro **i** em real. Que valores são armazenados em **a**, **b** e **n** após os seguintes comandos serem executados?

```
REAL :: a, b
INTEGER :: n, i, j
i = 10.
j = 3
n = i / j
a = i / j
b = REAL(i) / j
```

3 – Que valores serão exibidos após os seguintes comandos serem executados?

```
PROGRAM quiz_1
  INTEGER :: i
  REAL :: a
  a = 0.05
  i = nint(2.*3.141593/a)
  a = a * (5/3)
  WRITE(*,*) i, a
END PROGRAM quiz_1
```

4- Escreva uma `function` para calcular a distância entre dois pontos  $(x_1, y_1)$  e  $(x_2, y_2)$ .

5 – a) Escreva uma `function` que receba 3 valores **reais** e determine se eles podem representar os lados de um triângulo.

b) Escreva uma `function` que receba 3 valores **inteiros** e determine se eles podem representar os lados de um triângulo retângulo.

6 – Se conhecemos o logaritmo de um número **N** na base **b**, podemos calcular o seu logaritmo na base **a** através da fórmula.

$$\log_a(x) = \frac{\log_b(x)}{\log_b(a)}$$

Escreva uma função que calcule o logaritmo de um número **x** em uma base **a**, qualquer.

7 – Uma pessoa registrou o consumo de gasolina de seu carro, anotando os valores da quilometragem e da quantidade de litros de gasolina em várias enchidas do tanque. Escreva um programa que receba as quantidades de quilômetros e de litros em cada tanque cheio. O programa então deve calcular e exibir o consumo em quilômetros por litro para cada tanque cheio. Depois de processar toda a informação de entrada o programa deve calcular e exibir os quilômetros por litro combinados de todos os tanques cheios. A saída deve ser algo assim:

```
Quantos litros? (-1 para terminar)
40.5
Quantos quilômetros?
465.7
Consumo: 11.4987659 km/l

Quantos litros? (-1 para terminar)
38.7
Quantos quilômetros?
425.
Consumo: 10.9819117 km/l

Quantos litros? (-1 para terminar)
30.
Quantos quilômetros?
360.
Consumo: 12.0000000 km/l

Quantos litros? (-1 para terminar)
-1

O consumo total foi de 11.4532967 km/l
```

8 – Qualquer conjunto de três números inteiros que são lados de um triângulo retângulo é chamado de Trio Pitagórico. Encontre todos os trios pitagóricos formados por inteiros não maiores do que 500. Use laços de DO para testar todas as possibilidades. Este é um exemplo de computação de “força bruta”. Em cursos mais avançados você vai aprender que existem vários problemas interessantes para os quais não há nenhuma abordagem algorítmica conhecida a não ser usar pura força bruta.

9 – a) Escreva uma `function` para calcular a série de Taylor das funções seno e cosseno, com a máxima precisão possível. Teste seu programa para diferentes valores de argumentos das funções. Você irá verificar que seu programa começa a falhar para valores grandes dos argumentos. Explique por que isto acontece.

b) Modifique seu programa para que possa calcular os valores com boa precisão, para qualquer valor do argumento.

Dica: se o número  $x$  dado como argumento da função for maior do que  $\pi$  ou menor do que  $-\pi$ , calcule o número entre  $-\pi$  e  $\pi$  que tenha os mesmos valores de seno e cosseno que o  $x$ , e então aplique este número na série de Taylor.

10 – Escreva uma função para simular o lançamento de um dado, usando a sub-rotina implícita `RANDOM_NUMBER`. A função, do tipo `integer`, tem lista de argumentos vazia e sua saída é um número inteiro entre 1 e 6. Teste sua função para verificar se as seis faces tem a mesma probabilidade no lançamento do dado: execute a função muitas vezes e conte as ocorrências de cada número. Se seu dado não é “viciado” cada face tem que aparecer aproximadamente 1/6 das vezes.

11 – Neste exercício você vai usar sua function de lançamento de dado para programar uma versão simples do jogo chamado CRAPS.

*Craps* é um jogo de dados muito comum nos cassinos (desculpem, mas não sei o nome desse jogo em português). Nele um jogador joga contra a “casa”. As regras são as seguintes: O jogador lança dois dados. Se o resultado dos dados somar 7 ou 11 na primeira jogada, o jogador ganha. Se a soma for 2, 3 ou 12 na primeira jogada (“craps”) o jogador perde, ou seja, ganha a casa. Se a soma for 4, 5, 6, 8, 9 ou 10 na primeira jogada, então aquele valor se torna o “ponto” do jogador. Para ganhar, o jogador deve continuar lançando os dados até que seu ponto apareça novamente. O jogador perde se aparecer um 7 antes de aparecer seu ponto novamente.

a) Implemente as regras do jogo em um programa chamado *craps*, usando sua função de simulação do lançamento do dado. O programa deve terminar exibindo na tela a frase “O jogador ganhou!” ou “O jogador perde.”

b) Modifique o programa do jogo de dados para incluir apostas. Transforme o programa que joga uma partida de “craps” em uma function. Inicialize a variável `deposito` com 1000 reais. Peça ao jogador para fazer uma aposta. Use um laço `while` para verificar que a aposta é menor ou igual ao depósito e, se não for, peça ao jogador para apostar outro valor até um valor de aposta válido seja escrito. Feita a aposta, execute o jogo uma vez. Se o jogador ganhar, some ao depósito o valor da aposta e mostre o novo valor do depósito. Se o jogador perder, subtraia do depósito o valor da aposta, mostre o novo valor do depósito. Ao final de cada rodada verifique se o depósito se tornou zero e, caso isto aconteça, exiba a mensagem “Lamento, você faliu.”