



Programação de Computadores I BCC 701 2015-2
Simulado da Segunda Avaliação 17/12/2015 Valor (10,0) Duração: 110 minutos

ATENÇÃO: Leia com atenção as questões da prova. A interpretação do enunciado faz parte da avaliação. Todos os programas devem ser escritos em SciLab.

B O A P R O V A !

Aluno: _____

Matrícula: _____ Turma: _____

Assinatura: _____

Questão 1 (2.5)

O valor de π pode ser aproximado pelo somatório infinito:

$$\pi \approx 3 + \frac{4}{2 \times 3 \times 4} - \frac{4}{4 \times 5 \times 6} + \frac{4}{6 \times 7 \times 8} - \frac{4}{8 \times 9 \times 10} + \frac{4}{10 \times 11 \times 12} - \dots$$

Escreva um programa para calcular um valor aproximado para π de acordo com a fórmula acima. O programa deve ler o número ***n*** de parcelas a serem usadas no cálculo do somatório. O programa deve testar se valor de ***n*** é um inteiro positivo, solicitando que seja digitado novo valor de ***n***, repetidamente, enquanto o valor lido não for válido.

Dois exemplos de execução do programa são apresentados a seguir.

Execução 1

```
QUAL A QUANTIDADE DE PARCELAS (n)?: 0
ERRO: VALOR INVÁLIDO PARA n.
QUAL A QUANTIDADE DE PARCELAS (n)?: -1
ERRO: VALOR INVÁLIDO PARA n.
QUAL A QUANTIDADE DE PARCELAS (n)?: 2.2
ERRO: VALOR INVÁLIDO PARA n.
QUAL A QUANTIDADE DE PARCELAS (n)?: 1

VALOR APROXIMADO DE Pi = 3
```

Execução 2

```
QUAL A QUANTIDADE DE PARCELAS (n)?: 3

VALOR APROXIMADO DE Pi = 3.13333
```

Execução 2

```
QUAL A QUANTIDADE DE PARCELAS (n)?: 1000

VALOR APROXIMADO DE Pi = 3.14159
```

Proposta de Código para a Solução

```
clear; clc;
n = input("QUAL A QUANTIDADE DE PARCELAS (n)?: ")
while n < 0 | n <> int(n)
    printf("ERRO: VALOR INVÁLIDO PARA n.\n")
    n = input("QUAL A QUANTIDADE DE PARCELAS (n)?: ")
end

soma = 3;
sinal = 1;
for i=2:n
    soma = soma + ((4/(i*(i+1)*(i+2)))*sinal)
    sinal = sinal * -1
end
printf("\nVALOR APROXIMADO DE Pi = %g",soma)
```

Proposta de Critérios de Correção

Critério	Valor
TOTAL	2,5

Questão 2 (2.5)

Faça um programa que leia o número de alunos de uma turma (n) e, em seguida, leia as notas de cada um dos alunos e determine e imprima os seguintes dados:

- A média das notas dos alunos
- O número de alunos aprovados, isto é, com nota igual ou maior que 6,0.

OBS: O programa não precisa verificar se o valor lido para o número de alunos ou para cada uma das notas é válido, podendo supor que sempre será digitado um valor válido.

Um exemplo de execução é mostrado a seguir.

Execução

```
SISTEMA DE NOTAS
NÚMERO DE ALUNOS DA TURMA: 7

DIGITE A NOTA DO ALUNO 1: 8.5
DIGITE A NOTA DO ALUNO 2: 4.5
DIGITE A NOTA DO ALUNO 3: 7.3
DIGITE A NOTA DO ALUNO 4: 9.2
DIGITE A NOTA DO ALUNO 5: 6.4
DIGITE A NOTA DO ALUNO 6: 3.7
DIGITE A NOTA DO ALUNO 7: 7.8

MÉDIA DAS NOTAS = 6.77
NÚMERO DE ALUNOS APROVADOS = 5
```

Proposta de Código para a Solução

```
clear; clc;
printf("\nSISTEMA DE NOTAS")
pop = input("Número de alunos da turma: ")

soma = 0; naprov = 0
for i=1:n do
    nota = input(sprintf("Digite a nota do aluno %g: ",i))
    soma = soma + nota
    if nota >= 6.0 then
        naprov = naprov +1
    end
end
printf("\nMÉDIA DAS NOTAS DA TURMA = %4.2f",soma/n)
printf("\nNÚMERO DE ALUNOS APROVADOS = %2g",naprov)
```

Proposta de Critérios de Correção

Critério	Valor
TOTAL	2,5

Questão 3 (2.5)

O setor de compras do UFOP está realizando uma cotação de preços para comprar 10 novos computadores para o laboratório de computação. Você vai escrever um programa para auxiliar a UFOP nessa tarefa. O programa deve ler os seguintes dados:

1. O valor que a UFOP tem disponível para a compra de novos computadores.
2. O preço do computador, fornecido por cada uma das empresas nas quais foi feita a cotação de preços. A leitura desses valores deve terminar quando for digitado um valor menor ou igual a 0 (zero) para o preço do computador.

O programa deve imprimir os seguintes dados:

1. O preço mínimo, e o número da empresa que ofereceu o preço mínimo.
2. Caso o dinheiro disponível seja suficiente para comprar os 10 novos computadores (pelo preço mínimo), o programa deve informar qual será o saldo restante. Caso contrário deverá informar qual é o maior número de computadores que é possível adquirir com o dinheiro disponível e qual é o saldo restante.

OBS: Não é necessário validar os dados de entrada, isto é, supõe-se que sempre serão digitados valores válidos. A entrada e saída de dados deve ser conforme mostrado nos exemplos de execução a seguir.

Exemplo de Execução 1

Compra de Computadores

Digite o valor disponível: 15000.00

Preço do produto na loja 1(ou zero para encerrar): 2300.00

Preço do produto na loja 2(ou zero para encerrar): 1950.00

Preço do produto na loja 3(ou zero para encerrar): 1480.00

Preço do produto na loja 4(ou zero para encerrar): 990.00

Preço do produto na loja 6(ou zero para encerrar): 0

Menor preço = R\$ 990.00 (empresa 4)

Os computadores podem ser comprados

O saldo restante será de R\$ 5100.00

Exemplo de Execução 2

Compra de Computadores

Digite o valor disponível: 7000.00

Preço do computador na loja 1(ou zero para encerrar): 2300.00

Preço do computador na loja 2(ou zero para encerrar): 1950.00

Preço do computador na loja 3(ou zero para encerrar): 1480.00

Preço do computador na loja 4(ou zero para encerrar): 990.00

Preço do computador na loja 6(ou zero para encerrar): 0

Menor preço = R\$ 990.00 (loja 4)

Só podem ser comprados 7 computadores

O saldo restante será de R\$ 70.00

Proposta de Código para a Solução

```
clear; clc;
printf("\nCOMPRA DE COMPUTADORES\n")
disp = input("Digite o valor disponível: ")

loja = 1; pmin = %inf; lojamin = 0;
preco = input(sprintf("Preço do computador na loja %g :",loja))
while preco > 0 do
    if preco < pmin then
        pmin = preco; lojamin = loja
    end
    loja = loja + 1
end
printf("Menor preço = R$ %5.2f (loja %g)\n",pmin,lojamin)
ptot = pmin * 10
if ptot <= disp then
    printf("Os computadores podem ser comprados\n",ano)
    printf("O saldo restante será R$ %5.2f\n",disp-ptot)
else
    n = int(disp/pmin)
    printf("Só podem ser comprados %g computadores \n",n)
    printf("O saldo restante será R$ %5.2f\n",disp-n*pmin)
end
```

Proposta de Critérios de Correção

Critério	Valor
TOTAL	2,5

Questão 4 (2.5)

Dê uma definição para a função **maxdiv**, que recebe como parâmetro um número inteiro positivo **n** e retorna o maior divisor próprio de **n**. O maior divisor próprio de **n** é o maior divisor de **n** diferente dele próprio. Por exemplo, o maior divisor próprio de 12 é 6 e o maior divisor próprio de 15 é 5.

Escreva agora um programa que leia dois números inteiros positivos, imprima o maior divisor próprio de cada um deles e imprima a soma dos maiores divisores próprios desses dois números, conforme ilustra o exemplo de execução a seguir.

Execução

```
SOMA DOS MAIORES DIVISORES
```

```
DIGITE UM NÚMERO INTEIRO POSITIVO: 8
```

```
MAIOR DIVISOR PRÓPRIO = 4
```

```
DIGITE UM NÚMERO INTEIRO POSITIVO: 15
```

```
MAIOR DIVISOR PRÓPRIO = 5
```

```
SOMA DOS DIVISORES = 9
```

Proposta de Código para a Solução

```
clear; clc;
function res = maxdiv(n)
    divp = 2
    while modulo(n,res)<>0 do
        divp = divp+1
    end
    res = n/divp
endfunction

// Programa principal
printf("SOMA DOS MAIORES DIVISORES\n")
n1 = input("Digite um inteiro positivo: ")
d1 = maxdiv(n)
printf("Maior divisor próprio = %g\n",d1)

n2 = input("Digite um inteiro positivo: ")
d2 = maxdiv(n)
printf("Maior divisor próprio = %g\n",d2)

printf("SOMA DOS DIVISORES = %g",d1+d2)
```

Proposta de Critérios de Correção

Critério	Valor
TOTAL	2,5