
PROGRAMAÇÃO DE COMPUTADORES I – BCC701

CADERNO DE EXERCÍCIOS

MÓDULO 5 – ESTRUTURAS DE REPETIÇÃO

2020/1

ELABORADO PELA COMISSÃO DE UNIFICAÇÃO DA DISCIPLINA BCC701,
COM A COLABORAÇÃO DE PROFESSORES E ESTAGIÁRIOS DOCENTES

<http://www.decom.ufop.br/bcc701/>

DECOM – DEPARTAMENTO DE COMPUTAÇÃO
ICEB – INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
UFOP – UNIVERSIDADE FEDERAL DE OURO PRETO

Sumário

5	Estrutura de Repetição	2
Questão 5.1.	(2018-2)	2
Questão 5.2.	(2018-1)	3
Questão 5.3.	(2019-2)	4
Questão 5.4.	(2018-2)	4
Questão 5.5.	(2019-2)	6
Questão 5.6.	(2013-2)	7
Questão 5.7.	(2013-2)	7
Questão 5.8.	(2019-2)	8
Questão 5.9.	(2019-1)	9
Questão 5.10.	(2018-2)	9
Questão 5.11.	(2019-1)	10
Questão 5.12.	(2018-1)	11
Questão 5.13.	(2017-1)	12
Questão 5.14.	(2017-2)	13
Questão 5.15.	(2016-2)	15
Questão 5.16.	(2014-2)	15
Questão 5.17.	(2016-1)	16

Estrutura de Repetição

Questão 5.1 (2018-2)

O **Índice de Massa Corporal**, ou apenas **IMC**, é uma medida internacional que serve para definir se uma pessoa está em seu peso ideal, abaixo ou acima dele. Uma classificação simplificada do **IMC** é dada na tabela abaixo:

IMC	Classificação
$IMC < 16$	Magreza grave
$16 \leq IMC < 18,5$	Abaixo do peso
$18,5 \leq IMC < 25$	Saudável
$25 \leq IMC < 30$	Sobrepeso
$30 \leq IMC < 40$	Obesidade
$IMC \geq 40$	Obesidade extrema

Sabendo que o cálculo do IMC é dado pela fórmula $IMC = \text{peso} / (\text{altura} * \text{altura})$, fazer um programa para classificar a condição de um dado número de pacientes. O programa deve inicialmente ler o total de pacientes (inteiro) e depois, para cada paciente, ler o peso e altura (reais), calcular o seu respectivo IMC e imprimir na tela a sua classificação, de acordo com a tabela anterior. Ao final o programa deve informar o percentual de pacientes contidos em duas classes: “Magreza grave” e “Obesidade extrema”. Todos os valores calculados devem ser exibidos com formatação de 2 casas decimais.

Exemplo 1:

```
Quantidade de pacientes: 2
Peso: 100
Altura: 1.5
O IMC é 44.44 ==> Obesidade extrema
Peso: 50
```

```
Altura: 1.80
O IMC é 15.43 ==> Magreza grave
Percentual para Magreza grave: 50.00%.
Percentual para Obesidade extrema: 50.00%.
```

Exemplo 2:

```
Quantidade de pacientes: 3
Peso: 110
Altura: 1.65
O IMC é 40.40 ==> Obesidade extrema
Peso: 80
Altura: 1.7
O IMC é 27.68 ==> Sobrepeso
Peso: 70
Altura: 1.75
O IMC é 22.86 ==> Saudável
Percentual para Magreza grave: 0.00%.
Percentual para Obesidade extrema: 33.33%.
```

Questão 5.2 (2018-1)

Um bombeiro hidráulico deve fazer uma série de encanamentos, cada um de um dado comprimento total e para isso ele dispõe de canos do tipo C1 com 1,7m e do tipo C2 com 2,3m. Por experiência, para cada encanamento ele chuta uma quantidade de canos de cada tipo. Mas ele deseja que você faça um programa para verificar a exatidão dos chutes que ele dá. Desta forma, faça um programa que leia uma série de comprimentos de encanamento, e para cada encanamento, a quantidade de canos do tipo C1 e do tipo C2 e mostre se o bombeiro acertou na mosca, se sobrou cano ou se faltou cano. Caso tenha sobrado ou faltado, o programa deve informar a metragem total que irá sobrar ou faltar para completar o encanamento. O programa deve ser finalizado quando for digitado 0 para o comprimento do encanamento.

Exemplo 1:

```
Entre com o comprimento do encanamento (m) ou 0 para fim: 17.7
Total de canos tipo C1 (1.7m): 5
Total de canos tipo C2 (2.3m): 4
Acertou na mosca, não sobra nem falta cano!
Entre com o comprimento do encanamento (m) ou 0 para fim: 13.9
Total de canos tipo C1 (1.7m): 4
Total de canos tipo C2 (2.3m): 2
Vai faltar 2.5 metros de cano
Entre com o comprimento do encanamento (m) ou 0 para fim: 28
Total de canos tipo C1 (1.7m): 6
Total de canos tipo C2 (2.3m): 9
Vai sobrar 2.9 metros de cano
Entre com o comprimento do encanamento (m) ou 0 para fim: 0
```

Questão 5.3 (2019-2)

Você é o síndico do prédio onde mora. Haverá uma eleição para a escolha de uma chapa para compor a nova administração do prédio, composta de síndico, tesoureiro e secretário. Sua responsabilidade é apurar os votos e determinar a chapa vencedora. Considerando que são apenas três chapas, denominadas de “a”, “b” e “c”, você desenvolverá um programa para fazer a apuração. O programa funciona da seguinte maneira, é fornecida uma entrada para cada voto enquanto não for digitado o valor “q”, indicando que não há mais votos a serem computados. Cada voto digitado indica o nome da chapa (“a”, “b” ou “c”), qualquer voto diferente destes valores são considerados nulos. Após a entrada de todos os votos, o programa imprime o resultado de apuração dos votos conforme os exemplos a seguir.

Exemplo 1:

```
Chapa: a
Chapa: a
Chapa: b
Chapa: x
Chapa: q
Resultado:
- Chapa a: 2 voto(s)
- Chapa b: 1 voto(s)
- Chapa c: 0 voto(s)
- Nulos : 1 voto(s)
```

Exemplo 2:

```
Chapa: c
Chapa: a
Chapa: b
Chapa: x
Chapa: y
Chapa: b
Chapa: b
Chapa: c
Chapa: b
Chapa: q
Resultado:
- Chapa a: 1 voto(s)
- Chapa b: 4 voto(s)
- Chapa c: 2 voto(s)
- Nulos : 2 voto(s)
```

Questão 5.4 (2018-2)

O Campus Aberto da UFOP realizará uma gincana e precisa de um programa para computar a pontuação das duas equipes participantes e determinar a equipe vencedora da seguinte forma:

1. serão realizadas N provas, definidas por entrada do usuário;
2. cada equipe receberá uma nota entre 0 e 10 para cada prova (não é necessário validar);

3. a equipe com maior pontuação será a vencedora da prova, podendo haver empate;
4. para cada prova, deverá ser lida a nota de cada equipe e computada a equipe vencedora;
5. ao final informar qual foi a equipe que venceu o maior número de provas, ou se houve empate;
6. o programa imprime a pontuação de cada equipe e a equipe vencedora na tela.

Exemplo 1:

```
Quantidade de provas: 5
=== Prova 1 ===
Pontuação da equipe 1: 5
Pontuação da equipe 2: 6
=== Prova 2 ===
Pontuação da equipe 1: 7.8
Pontuação da equipe 2: 7.6
=== Prova 3 ===
Pontuação da equipe 1: 5
Pontuação da equipe 2: 4
=== Prova 4 ===
Pontuação da equipe 1: 6
Pontuação da equipe 2: 7
=== Prova 5 ===
Pontuação da equipe 1: 3
Pontuação da equipe 2: 4
=== Resultados ===
A equipe 1 venceu 2 prova(s)
A equipe 2 venceu 3 prova(s)
Houve 0 empate(s) entre as equipes
Equipe 2 é a vencedora
```

Exemplo 2:

```
Quantidade de provas: 3
=== Prova 1 ===
Pontuação da equipe 1: 5
Pontuação da equipe 2: 5
=== Prova 2 ===
Pontuação da equipe 1: 8
Pontuação da equipe 2: 6
=== Prova 3 ===
Pontuação da equipe 1: 4
Pontuação da equipe 2: 6
=== Resultados ===
A equipe 1 venceu 1 prova(s)
A equipe 2 venceu 1 prova(s)
Houve 1 empate(s) entre as equipes
Houve empate entre as equipes!
```

Questão 5.5 (2019-2)

O preparador físico de um atleta de maratona deseja calcular o tempo estimado para completar uma prova com base no tempo gasto para percorrer uma pequena parte do percurso. Para isso ele recorre a uma formulação matemática definida a seguir.

Considerando o percurso **AB**, que determina o percurso total da maratona, o trecho **AB'**, que determina um pequeno trecho deste percurso, e **t** o tempo para percorrer o trecho **AB'**, o cálculo do tempo total **T**, estimado para percorrer o percurso total **AB** com base no tempo **t**, com uma precisão de **n** termos, é dado pelo somatório:

$$T = \frac{t}{2^0} + \frac{t}{2^1} + \frac{t}{2^2} + \dots + \frac{t}{2^{n-1}}$$

Ajude o preparador físico implementando um programa que receba como entradas os valores de **t** e **n** (ambos inteiros), calcule e imprima o valor de **T**. Observe que a saída é formatada com 4 casas decimais.

Exemplo 1:

```
Entre com t: 10
Entre com n: 5
O tempo estimado é 19.3750
```

Exemplo 2:

```
Entre com t: 10
Entre com n: 10
O tempo estimado é 19.9805
```

Exemplo 3:

```
Entre com t: 10
Entre com n: 15
O tempo estimado é 19.9994
```

Exemplo 4:

```
Entre com t: 10
Entre com n: 20
O tempo estimado é 20.0000
```

Exemplo 5:

```
Entre com t: 10
Entre com n: 25
O tempo estimado é 20.0000
```

Questão 5.6 (2013-2)

Uma *Sequência de Collatz* modificada pode ser definida do seguinte modo:

- Dado um número inteiro positivo n , se o resto da divisão inteira de n por 3 for 0, divida n por 3 ($n/3$);
- se o resto for 1, multiplique n por 4, some 2 e divida o resultado por 3 ($(4n+2)/3$);
- se o resto for 2, multiplique n por 2, subtraia 1 e divida o resultado por 3 ($(2n-1)/3$).
- Repita esse processo para o valor obtido, e assim sucessivamente, até que o valor obtido seja igual a 1.

Escreva um programa que leia um valor inteiro positivo n e imprima os valores da *Sequência de Collatz* para n . Não é necessário verificar se o valor digitado é válido.

Exemplo 1:

```
Digite um número inteiro positivo: 12
Sequencia de Collatz:
12 4 6 2 1
```

Exemplo 2:

```
Digite um número inteiro positivo: 231
Sequencia de Collatz:
231 77 51 17 11 7 10 14 9 3 1
```

Questão 5.7 (2013-2)

A série de *RICCI* é gerada da seguinte forma:

- os dois primeiros termos são fornecidos pelo usuário;
- os demais termos desta série são gerados como sendo a soma dos 2 termos anteriores.

Escreva um programa que solicite ao usuário os dois primeiros termos (números inteiros). Os valores informados devem ser positivos e o programa deve obrigar que o usuário os informe corretamente. A seguir o programa deve imprimir todos os termos da série de *RICCI* que sejam menores que o quadrado do produto dos dois primeiros termos, fornecidos pelo usuário.

Exemplo 1:

```
Digite o primeiro termo da série: -3
Digite o segundo termo da série: 4

Os valores informados devem ser positivos. Digite novamente.
Digite o primeiro termo da série: 3
Digite o segundo termo da série: 4

Série de RICCI:
3 4 7 11 18 29 47 76 123
```


Questão 5.8 (2019-2)

As regras de transição para aposentadoria estão causando muita dúvida e seus amigos pediram uma ajuda. Todos eles se encaixam na regra definida a seguir:

- **Regra de transição por pontos - Setor Privado:** soma-se o tempo de contribuição com a idade, caso atinja a pontuação mínima exigida para o ano em questão, pode se aposentar. O cálculo começa no ano de 2019, e a pontuação mínima neste ano é de 96 pontos para os homens e 86 pontos para as mulheres, e é acrescida de um a cada ano, atingindo o máximo de 105 para homens e 100 para mulheres.

Implemente um programa que realize a simulação do cálculo do ano no qual um amigo poderá se aposentar. As entradas para o programa são: **sexo** (1 - feminino e 2 - masculino), **tempo de contribuição em 2019** e a **idade em 2019**. O programa imprime como resultado uma tabela com o ano (começando de 2019), a pontuação mínima para o ano e a pontuação obtida no ano, encerrando quando a pontuação obtida no ano for igual ou superior à pontuação mínima para este mesmo ano. O amigo, em nenhum momento, interrompe sua contribuição, ou seja, assim como sua idade, o tempo de contribuição é acrescido de um a cada ano que se passa. Observe que a pontuação obtida inicialmente é a idade mais o tempo de contribuição (ambos em 2019), e para cada ano seguinte a pontuação obtida é acrescida de 2 (1 pela idade e 1 pela contribuição).

Exemplo 1:

```
Sexo: 2
Tempo de contribuição em 2019: 20
Idade em 2019: 45
Ano | Pont. Min. | Pont. Obtida
2019 |          96 |          65
2020 |          97 |          67
2021 |          98 |          69
2022 |          99 |          71
2023 |         100 |          73
2024 |         101 |          75
2025 |         102 |          77
2026 |         103 |          79
2027 |         104 |          81
2028 |         105 |          83
2029 |         105 |          85
2030 |         105 |          87
2031 |         105 |          89
2032 |         105 |          91
2033 |         105 |          93
2034 |         105 |          95
2035 |         105 |          97
2036 |         105 |          99
2037 |         105 |         101
2038 |         105 |         103
2039 |         105 |         105
```

Questão 5.9 (2019-1)

Uma empresa aplicará um reajuste em seus produtos baseado no tipo e preço, conforme a tabela:

Tipo	Preço atual (P)		
	P ≤ R\$ 50,00	R\$ 50,00 < P ≤ R\$ 100,00	P > R\$ 100,00
Alimentos	10%	12%	15%
Limpeza	9%	10%	11%
Transporte	6%		
“outros”	4%		

Implemente um programa que receba como entradas os dados de vários produtos: o tipo (texto) e o preço atual (número real); em seguida, calcula e imprime o preço reajustado. O preço atual não pode ser inferior a 0, caso isso ocorra, a mensagem “Preço inválido!” deve ser impressa na tela. O programa deve ser encerrado quando o usuário digitar “fim” para o tipo do produto. Observe que os valores numéricos são impressos com duas casas decimais de precisão e que “outros” se refere a valores que não estejam representados na tabela (qualquer coisa diferente de “Alimentos”, “Limpeza” e “Transporte” – incluindo também “fim”, usado para encerrar o programa).

exemplo 1

```
Tipo: Limpeza
Preço atual: 40
Preço reajustado: R$ 43.60
Tipo: Limpeza
Preço atual: 140
Preço reajustado: R$ 155.40
Tipo: Transporte
Preço atual: 0
Preço reajustado: R$ 0.00
Tipo: Transporte
Preço atual: -23
Preço Inválido!
Preço atual: 150
Preço reajustado: R$ 159.00
Tipo: Entretenimento
Preço atual: 100
Preço fim
```

Questão 5.10 (2018-2)

Um estudante de Biologia quer “simular” uma série de crescimentos populacionais. Neste estudo, temos cada espécie uma população inicial N_0 , uma taxa de crescimento r , que corresponde à taxa de natalidade da espécie e o seu nível de saturação K , que corresponde ao número máximo de indivíduos que o ambiente comporta. O estudo consiste em simular o crescimento de uma série de espécies distintas, sendo que para cada espécie são conhecidas as seguintes informações:

1. a população inicial N_0 ;
2. a taxa de crescimento da população r ; e

3. o nível de saturação K .

Sabendo-se que o crescimento populacional é dado pela fórmula: $N_0 = N_{i-1} * (1 + r)$, ou seja, o número de indivíduo da próxima geração (i) é igual ao número de indivíduos da geração atual ($n-1$) acrescido dos “novos” indivíduos, escreva um programa para:

1. ler total de espécies do estudo (inteiro);
2. para cada espécie, fazer:
 - (a) ler a população inicial N_0 (inteiro);
 - (b) ler a taxa de crescimento da população r (real); e
 - (c) ler o nível de saturação K (inteiro).
 - (d) Calcular e informar quantas gerações são necessárias para que a população de cada espécie ultrapasse o nível de saturação K e a população final.

Exemplo 1:

```
Número de espécies: 3
Entre com os dados da Espécie 1
- População Inicial: 100
- Taxa de crescimento: 0.3
- Nível de saturação: 500
- São necessárias 7 gerações.
- População final: 627
Entre com os dados da Espécie 2
- População Inicial: 10
- Taxa de crescimento: 0.05
- Nível de saturação: 100
- São necessárias 48 gerações.
- População final: 104
Entre com os dados da Espécie 3
- População Inicial: 50
- Taxa de crescimento: 0.03
- Nível de saturação: 500
- São necessárias 78 gerações.
- População final: 501
```

Questão 5.11 (2019-1)

Barba Negra e Anne Bonny são piratas disputando um tesouro. Visando evitar mais mortes entre seus tripulantes eles combinaram de disputar um jogo de adivinhação. Barba Negra escolherá um número inteiro E entre 1 e N (sendo N um número inteiro definido pela Anne Bonny). Anne Bonny terá no máximo X tentativas para acertar o número (sendo X outro número inteiro definido por Barba Negra). A cada tentativa de Anne Bonny, Barba Negra deverá fornecer um novo intervalo de valores inteiros, $[A, B]$, ou encerrar o jogo da seguinte maneira:

1. Inicialmente o intervalo de valores é $[1; N]$, ou seja, $A = 1$ e $B = N$;
2. Considerando que a tentativa de Anne Bonny seja o valor T , A e B serão atualizados pela regra:

- (a) caso T seja menor do que E , o valor de A muda para T , e B não muda;
 - (b) caso T seja maior do que E , o valor de B muda para T , e A não muda;
3. O jogo termina quando $T = E$ ou Anne Bonny atingir o número máximo de tentativas (X);
4. Anne Bonny ganha se $T = E$, caso contrário Barba Negra ganha.

Escreva um programa que fará o papel de juiz desta disputa entre Barba Negra e Anne Bonny através de entradas do usuário.

Exemplo 1:

```
Anne Bonny, defina N: 50
Barba Negra, defina E: 30
Barba Negra, defina X: 3
Anne Bonny, escolha um número entre 1 e 50: 40
Anne Bonny, escolha um número entre 1 e 40: 20
Anne Bonny, escolha um número entre 20 e 40: 25
Anne Bonny você consumiu suas 3 tentativas, Barba Negra ganhou!
```

Exemplo 2:

```
Anne Bonny, defina N: 50
Barba Negra, defina E: 30
Barba Negra, defina X: 3
Anne Bonny, escolha um número entre 1 e 50]: 28
Anne Bonny, escolha um número entre 28 e 50]: 30
Anne Bonny você ganhou, o número é 30!
```

Exemplo 3:

```
Anne Bonny, defina N: 100
Barba Negra, defina E: 60
Barba Negra, defina X: 3
Anne Bonny, escolha um número entre 1 e 100]: 23
Anne Bonny, escolha um número entre 23 e 100]: 78
Anne Bonny, escolha um número entre 23 e 78]: 60
Anne Bonny você ganhou, o número é 60!
```

Questão 5.12 (2018-1)

Um aluno da UFOP é responsável pelas compras mensais de supermercado de uma quantidade determinada de repúblicas de estudantes. Para isto, ele recebe de cada uma das repúblicas a lista de produtos que serão comprados.

Escreva um programa que simula as compras das repúblicas. O programa deve calcular o valor da compra de cada uma das repúblicas e o valor total de todas as compras efetuadas de acordo com o exemplo a seguir.

Exemplo 1:

Informe a quantidade de repúblicas: 3

República número 1:

Quantos produtos na lista de compras: 2

Qual o valor do produto ? 15.00

Qual o valor do produto ? 12.00

Compras da República 1: R\$ 27.00

República número 2:

Quantos produtos na lista de compras: 4

Qual o valor do produto ? 10.00

Qual o valor do produto ? 20.00

Qual o valor do produto ? 30.00

Qual o valor do produto ? 40.00

Compras da República 2: R\$ 100.00

República número 3:

Quantos produtos na lista de compras: 5

Qual o valor do produto ? 1.00

Qual o valor do produto ? 2.00

Qual o valor do produto ? 3.00

Qual o valor do produto ? 4.00

Qual o valor do produto ? 5.00

Compras da República 3: R\$ 15.00

Total das compras: R\$ 142.00

Questão 5.13 (2017-1)

Em uma determinada competição de saltos ornamentais, cada salto recebe um grau de dificuldade e é avaliado por juízes. Após cada salto, os juízes, que não se comunicam uns com os outros, mostram suas notas. Um salto é pontuado entre zero e dez pontos. Depois de apresentadas as notas, a mais alta e a mais baixa são descartadas. O restante é somado e multiplicado pelo grau de dificuldade do salto, que pode estar entre 1.2 e 3.8, definido sempre antes do início da apresentação do atleta. O julgamento então é feito da seguinte forma. Suponha que um saltador tenha seu grau de dificuldade de movimento avaliado em 2.0 e tire as notas 6.0, 5.0, 5.0, 5.0, 5.0, 5.0, 4.0, de sete juízes. Retiram-se a nota mais baixa (4.0) e a nota mais alta (6.0), o que resulta na soma de notas igual a 25.0. A nota final é obtida multiplicando-se a soma de notas (25.0) pelo grau de dificuldade de movimento (2.0), resultando, neste exemplo, na nota 50.0. Implemente um programa que apure a pontuação dos competidores de acordo com estas regras, conforme o exemplo a seguir.

Exemplo 1:

Saltos Ornamentais:

Informe o número de competidores: 3

Informe o número de juízes: 7

Nome do competidor: Gabriela
Grau de dificuldade: 2.0
Nota juiz: 6.0
Nota juiz: 5.0
Nota juiz: 5.0
Nota juiz: 5.0
Nota juiz: 5.0
Nota juiz: 5.0
Nota juiz: 4.0
A Gabriela obteve 50.00 pontos.

Nome do competidor: Marina
Grau de dificuldade: 1.5
Nota juiz: 8.5
Nota juiz: 7.0
Nota juiz: 8.0
Nota juiz: 8.0
Nota juiz: 8.4
Nota juiz: 7.5
Nota juiz: 7.7
A Marina obteve 59.40 pontos.

Nome do competidor: Mafalda
Grau de dificuldade: 3.0
Nota juiz: 6.0
Nota juiz: 7.0
Nota juiz: 6.5
Nota juiz: 6.8
Nota juiz: 7.9
Nota juiz: 6.2
Nota juiz: 6.6
A Mafalda obteve 50.00 pontos.

Questão 5.14 (2017-2)

Implemente um programa para calcular a nota média dos alunos e verificar se eles possuem nota para **reprovação** (< 3), **exame especial** (≥ 3 e < 6) ou **aprovação** (≥ 6). O cálculo para a nota média é feito a partir de uma **média ponderada**, ou seja, cada avaliação possui um peso, e o cálculo é feito pela equação:

$$\text{Média} = \frac{\sum_{i=1}^{i=n} \text{Nota}_i * \text{Peso}_i}{\sum_{i=1}^{i=n} \text{Peso}_i}$$

O número de avaliações é igual para todos os alunos, mas os pesos podem variar. O programa deve inicialmente perguntar quantas avaliações serão aplicadas, em seguida, solicita os dados de cada aluno e imprime seu resultado. O programa continua sua execução enquanto o nome fornecido pelo usuário for diferente de vazio (“”), conforme o exemplo de execução abaixo.

Exemplo 1:

Informe o número de avaliações: 4

Nome do(a) aluno(a): Janaina

Nota 1: 7.4

Peso 1: 1

Nota 2: 6.8

Peso 2: 2

Nota 3: 7.3

Peso 3: 2

Nota 4: 6.7

Peso 4: 3

Janaina está aprovado(a) com média 6.96.

Nome do(a) aluno(a): Joaquim

Nota 1: 7.8

Peso 1: 2

Nota 2: 8.4

Peso 2: 1

Nota 3: 8.6

Peso 3: 1

Nota 4: 7.5

Peso 4: 2

Joaquim está aprovado(a) com média = 7.93.

Nome do(a) aluno(a): João

Nota 1: 8.7

Peso 1: 3

Nota 2: 7.5

Peso 2: 1

Nota 3: 1.5

Peso 3: 4

Nota 4: 2.4

Peso 4: 3

João está em exame especial com média 4.25.

Nome do(a) aluno(a):

Questão 5.15 (2016-2)

Escreva um programa que leia dois números inteiros positivos, **n1** e **n2**, e imprima uma figura tal como a mostrada a seguir, onde **n1** é o número de linhas e **n2** é o número de colunas da figura. No exemplo abaixo, **n1 = 5** e **n2 = 8**:

```
1 2 3 4 5 6 7 8
8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8
8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8
```

Caso seja informado número inferior a zero para **n1** ou **n2**, a seguinte mensagem deve ser impressa: “Dados de entrada inválidos”.

Exemplo 1:

```
Digite o número de linhas: 6
Digite o número de colunas: 5

1 2 3 4 5
5 4 3 2 1
1 2 3 4 5
5 4 3 2 1
1 2 3 4 5
5 4 3 2 1
```

Exemplo 2:

```
Digite o número de linhas: -1
Digite o número de colunas: 10

Dados de entrada inválidos
```

Questão 5.16 (2014-2)

Bart Simpson está aprendendo a jogar xadrez e tem dificuldade em saber para qual direção ele pode mover sua Torre.

Sabemos que um tabuleiro de xadrez é composto por **8 linhas** e **8 colunas**, e que a Torre se move ortogonalmente, ou seja, pelas linhas (horizontais) e pelas colunas (verticais).

Escreva um programa que solicite ao *Bart* a entrada do número da linha e da coluna que indicam a posição de sua Torre (ambos inteiros). O programa imprime quais são os possíveis movimentos da Torre. Utilize o valor **0** para indicar uma casa do tabuleiro para a qual a Torre **não pode ser movida** e o valor **1** para indicar uma casa para a qual ela **pode ser movida**.

Caso a linha ou a coluna não esteja no intervalo de números válidos, [**1**, **8**], a mensagem “Entrada inválida” deve ser impressa na tela e o programa deve solicitar um novo valor.

Exemplo 1:

```
Informe a linha: -6
Valor inválido, deve estar no intervalo [1, 8].
```



```
Informe a linha: 6
Informe a coluna: 4
```

Movimentos possíveis:

```
      1  2  3  4  5  6  7  8
-----
1 |  0  0  0  1  0  0  0  0
2 |  0  0  0  1  0  0  0  0
3 |  0  0  0  1  0  0  0  0
4 |  0  0  0  1  0  0  0  0
5 |  0  0  0  1  0  0  0  0
6 |  1  1  1  1  1  1  1  1
7 |  0  0  0  1  0  0  0  0
8 |  0  0  0  1  0  0  0  0
```

Exemplo 2:

```
Informe a linha: 2
Informe a coluna: 10
Valor inválido, deve estar no intervalo [1, 8].
Informe a coluna: 7
```

Movimentos possíveis:

```
      1  2  3  4  5  6  7  8
-----
1 |  0  0  0  0  0  0  1  0
2 |  1  1  1  1  1  1  1  1
3 |  0  0  0  0  0  0  1  0
4 |  0  0  0  0  0  0  1  0
5 |  0  0  0  0  0  0  1  0
6 |  0  0  0  0  0  0  1  0
7 |  0  0  0  0  0  0  1  0
8 |  0  0  0  0  0  0  1  0
```

Questão 5.17 (2016-1)

Escreva um programa que leia um valor inteiro positivo representando a dimensão do tabuleiro ($1 \leq n < 100$) e imprima um tabuleiro de jogo tal como ilustrado a seguir, para $n = 12$:

```
      1  2  3  4  5  6  7  8  9 10 11 12
1 x
2 # +
3 x x x
4 # + # +
5 x x x x x
6 # + # + # +
```

```

7 x x x x x x x
8 # + # + # + # +
9 x x x x x x x x
10 # + # + # + # + # +
11 x x x x x x x x x x
12 # + # + # + # + # + # +

```

Ou seja, as posições **abaixo da diagonal principal, incluindo essa diagonal**, são marcadas com **x**, nas linhas **ímpares**, ou com **#** e **+** alternados, nas linhas **pares**.

Caso o valor de **n** seja inválido (ou seja, não atenda $1 \leq n < 100$), a mensagem “Entrada inválida” deve ser impressa na tela e o programa finalizado.

Exemplo 1:

```

Dimensão do tabuleiro: 6
 1 2 3 4 5 6
1 x
2 # +
3 x x x
4 # + # +
5 x x x x x
6 # + # + # +

```

Exemplo 2:

```

Dimensão do tabuleiro: 7
 1 2 3 4 5 6 7
1 x
2 # +
3 x x x
4 # + # +
5 x x x x x
6 # + # + # +
7 x x x x x x x

```

Exemplo 2:

```

Dimensão do tabuleiro: 100
Entrada inválida

```