



# XI Olimpíada Cearense de Informática

2ª FASE - 31 de Outubro de 2023

## MODALIDADE PROGRAMAÇÃO

### **Leia atentamente as instruções:**

- Não serão permitidos empréstimos de materiais, consultas e comunicação entre os candidatos, tampouco o uso de livros e apontamentos. Relógios e aparelhos eletrônicos em geral deverão ser desligados. O não cumprimento destas exigências ocasionará a exclusão do candidato deste Exame;
- Aguarde o Aplicador da Prova autorizar a abertura do Caderno de Prova. Após a autorização, confira todas as questões antes de iniciar o Exame;
- Este Caderno de Prova contém 20 (vinte) questões objetivas, cada qual com apenas 1 (uma) alternativa correta;
- Não serão permitidas perguntas ao Aplicador da Prova sobre as questões da Prova;
- A duração desta prova será de 4 (quatro) horas;
- O tempo mínimo para ausentar-se definitivamente da sala é de 1 (uma) hora;
- Ao concluir a prova, permaneça em seu lugar e comunique ao Aplicador de Prova, sinalizando com uma de suas mãos;
- Aguarde autorização para devolver o Caderno de Prova.

## 2 | XI Olimpíada Cearense de Informática

**Questão 1.** Uma lenda conta a história de um sultão chamado Ladava, que desafiou um sábio matemático chamado Lahur Sessa para uma partida de xadrez. A lenda conta que o sultão ofereceu a seu oponente qualquer recompensa que ele quisesse, caso o sábio ganhasse a partida.

- Lahur Sessa venceu a partida, e fez um pedido aparentemente simples:
- Ele pediu um grão de trigo para a primeira casa do tabuleiro de xadrez, dois grãos para a segunda casa, quatro grãos para a terceira casa e assim por diante, dobrando a quantidade de grãos para cada casa sucessiva do tabuleiro, até a última casa.

Sabendo que há 64 casas num tabuleiro de xadrez, qual programa a seguir calcula corretamente a soma desejada?

(A) 01. Principal

```
02. inteiro: num, soma
03. num ← 0
04. soma ← 0
05. enquanto num <= 64 faça:
06.     soma ← soma + 2**num
07.     num ← num + 1
08. imprima(soma)
```

(B) 01. Principal

```
02. inteiro: num, soma
03. num ← 0
04. soma ← 0
05. enquanto num <= 63 faça:
06.     soma ← soma + 2**num
07.     num ← num + 1
08. imprima(soma)
```

(C) 01. Principal

```
02. inteiro num, soma
03. num ← 0
04. soma ← 0
05. enquanto num <= 63 faça:
06.     soma ← 2**num
07.     num ← num + 1
08. imprima(soma)
```

(D) 01. Principal

```
02. inteiro num, soma
03. num ← 0
04. soma ← 0
05. enquanto num <= 64 faça:
06.     soma ← 2**num
07.     num ← num + 1
08. imprima(soma)
```

### 3 | XI Olimpíada Cearense de Informática

(E) 01. Principal  
02. inteiro num, soma  
03. num ← 0  
04. soma ← 0  
05. enquanto num < 63 faça:  
06.       soma ← soma + 2\*\*num  
07.       num ← num + 1  
08. imprima(soma)

---

**Questão 2.** Analise o seguinte algoritmo:

```
1.       lista ← recebe uma lista de números
2.       PARA CADA i EM lista FAÇA
3.           op1 ← i * i * i
4.           op2 ← i - (i * 3)
5.           resultado ← op1 / op2
6.       ESCREVA resultado
7.       FIM PARA
```

Assinale a alternativa referente a qual o valor da variável “resultado” na 4ª execução do laço.

Considere que “lista” receba a sequência de números: 2, 5, 3, 1, 7, 4, 9?

- (A) - 0,5.
  - (B) 3.
  - (C) 0,5.
  - (D) -2.
  - (E) -1
- 

**Questão 3.** Analise a execução do algoritmo abaixo:

**Algoritmo FazAlgo**

Leia o valor de n

resultado ← 1

**para** i indo de 1 até n **faça**

    resultado ← resultado \* i

**se** resultado == i **faça**

        resultado = resultado \* resultado

**fim do loop**

**escreva** resultado

**Fim do algoritmo**

## 4 | XI Olimpíada Cearense de Informática

Se  $n = 5$ , qual o resultado da execução desse algoritmo?

- (A) 48
  - (B) 60
  - (C) 120
  - (D) 240
  - (E) 960
- 

**Questão 4.** Rebeca está fazendo um curso de programação e em uma das aulas se deparou com um algoritmo totalmente diferente do que ela havia estudado até então. Sendo uma garota super curiosa e observando que seus colegas ainda não haviam conseguido entender o funcionamento do algoritmo, ela tentou reescrevê-lo de outra forma e testou alguns valores em  $n$  para verificar seus resultados. O algoritmo que Rebeca viu na aula foi o seguinte:

**Algoritmo: Super(int n):**

1. int resultado = 1
2. se  $(n < 2)$  então
3.     retorne resultado
4. se  $(n \% 2 = 0)$  então
5.     retorne resultado + Super( $n-1$ )
6. senão
7.     retorne Super( $n-1$ )

Para garantir que o novo algoritmo retornasse resultados iguais ao primeiro, Rebeca escreveu quatro códigos e rodou cada um deles para valores de 0 à 100, mas apenas um estava correto em todos os casos. Dados os quatro algoritmos descritos abaixo, qual retorna o mesmo resultado que o **Super(int n)** nos casos testados por Rebeca?

(A) **Algoritmo: Mega(int n):**

1. int resultado = 0
2. para  $i \leftarrow 2$  até  $n$  faça
3.     se  $(i \% 2 = 0)$  então
4.         resultado = resultado +  $i$
5.     retorne resultado

(B) **Algoritmo: Giga(int n):**

1. int resultado = 0
2. enquanto  $(n > 0)$  faça
3.     se  $(n \% 2 = 1)$  então
4.         resultado = resultado + 1
5.      $n = n - 1$
6.     retorne resultado

## 5 | XI Olimpíada Cearense de Informática

(C) **Algoritmo: Hiper(int n):**

1. se  $(n < 2)$  então
2.     retorne 1
3. senão
4.     retorne  $(n // 2) + 1$

(D) **Algoritmo: Ultra(int n):**

1. resultado = 1
2. para  $i \leftarrow 1$  até  $n$  faça
3.     se  $(i \% 2 = 0)$  então
4.         resultado = resultado +  $i * 2$
5.     se  $(i + 1 > n)$  então
6.         resultado = resultado -  $i * 2$
7.     senão
8.         resultado = resultado -  $(i - 1) * 2$
9. retorne resultado

(E) Nenhuma das opções anteriores.

---

**Questão 5.** Uma das características mais importantes de uma linguagem de programação é seu grau de abstração, em outras palavras, o quão próximo a linguagem está do humano (alto nível) ou da máquina (baixo nível). Com base nisso assinale a alternativa correta:

- (A) A linguagem Java foi criada na década de 90 e possui como principais características ser uma linguagem de alto nível, orientada a objetos e com excelente portabilidade.
  - (B) É mais fácil aprender linguagens de programação de baixo nível porque são mais próximas da máquina, por isso os programas criados com essas linguagens funcionam melhor.
  - (C) O Assembly é uma das linguagens de máquina mais conhecidas e usadas atualmente, sendo considerada também uma linguagem de baixo nível.
  - (D) Python possui uma sintaxe simples de ser entendida e não são necessárias muitas linhas de código para escrever um programa nessa linguagem, por exemplo, um programa que escreve "Hello World!" no terminal é somente `print("Hello World!")`, por isso é considerada uma linguagem de baixo nível, já que não necessita de tanta experiência para ser aprendida.
  - (E) HTML é uma linguagem de programação eficaz na composição de sites. É organizada em tags de abertura e de fechamento, como o `<p>` que serve para criar um parágrafo e o `</p>` que fecha o parágrafo.
- 

**Questão 6.** Carlos Henrique desafiou sua namorada a descobrir seu apelido de infância a partir da interpretação de um algoritmo, o qual tem como entrada a string\* "Carlos\_Henrique".

\*String é um conjunto de caracteres armazenados em um vetor.

## 6 | XI Olimpíada Cearense de Informática

### Alg: Desafio\_Do\_Crush

#### Variáveis:

i: inteiro

Nome: vetor [15] de caractere;

Apelido: vetor [5] de caractere;

#### Início:

**para** i de 0 até 14 **faça**:

    escreva: "Digite uma letra do nome: ";

    leia: Nome[i];

**fimpara**;

**para** i de 0 até 1 **faça**:

    Apelido[i] <- Nome[i];

**fimpara**;

**para** i de 12 até 14 **faça**:

    Apelido[i-10] <- Nome[i];

**fimpara**;

    escreva: Apelido[];

**Fim**;

A partir da análise do algoritmo apresentado anteriormente, assinale abaixo o item que contém o apelido de infância de Carlos Henrique:

- (A) Caue.
- (B) Caque.
- (C) Carue.
- (D) Carque.
- (E) Cue.

---

**Questão 7.** Quando se fala de programação, existem vários tipos diferentes de linguagens que podem ser utilizadas. Quase toda linguagem é formulada com um propósito em mente, tendo suas vantagens, desvantagens e peculiaridades, além, é claro, de uma sintaxe própria. Quando se fala de velocidade e de consumo de memória, algumas são mais eficientes do que outras, com um dos exemplos mais famosos de eficiência sendo a linguagem de programação chamada C. Além de ser eficiente, C é considerada de baixo ou médio nível. Isso significa que sua sintaxe, apesar de não ser tão complicada e se distanciar da linguagem utilizada pelo computador, ainda sim tem suas complexidades.

## 7 | XI Olimpíada Cearense de Informática

Para o desenvolvimento de um aplicativo mobile, quando se vai escolher qual linguagem de programação utilizar, vários fatores são levados em consideração. Por causa de sua eficiência no controle de memória, do fato dela ser multiplataforma, de sua velocidade, e de sua escrita não ser tão complexa, uma das variações da linguagem C, a C# é frequentemente escolhida. Tendo como base as informações acima, qual das opções abaixo seria recomendável para desenvolvimento mobile?

\*\* Lembre-se que linguagens interpretadas tendem a ser mais lentas do que linguagens compiladas e que linguagens de alto nível são, frequentemente, mais fáceis de serem escritas e utilizadas do que as de baixo nível.

- (A) Python, uma linguagem interpretada acessível de alto nível, mas que não possui um bom controle de memória.
  - (B) COBOL, uma linguagem compilada complexa de baixo nível extremamente eficiente e veloz.
  - (C) Javascript, uma linguagem lenta interpretada de alto nível que usa bastante memória.
  - (D) Rust, uma linguagem multiplataforma compilada de sintaxe razoavelmente complicada, mas de alto nível, e bom controle de memória.
  - (E) Ruby, uma linguagem multiplataforma interpretada de alto nível não tão eficiente.
- 

**Questão 8.** Com base no seguinte algoritmo, analise-o e marque a opção que representa o que será retornado ao final de cada execução, tendo como entrada os seguintes valores, respectivamente:

Entrada 1: 25  
Entrada 2: 33  
Entrada 3: 71  
Entrada 4: 62

1. `entrada <- Leia()`
- 2.
3. `entrada <- entrada * 3`
- 4.
5. `se (entrada % 2) == 0:`
6.     `escreva("Caso 1")`
7. `senão se ((entrada * 3) % 5) == 0:`
8.     `escreva("Caso 2")`
9. `senão:`
10.    `escreva("Caso 3")`

## 8 | XI Olimpíada Cearense de Informática

Descrição:

' <- ' = Atribui um valor a uma variável

' \* ' = Retorna a multiplicação de dois números

' == ' = Retorna um valor booleano referente à comparação de dois valores

' + ' = Retorna a soma de dois números

' % ' = Retorna o valor do resto de uma divisão inteira entre dois números

(A) Caso 3, Caso 1, Caso 2, Caso 3

(B) Caso 2, Caso 3, Caso 2, Caso 1

(C) Caso 1, Caso 2, Caso 3, Caso 2

(D) Caso 2, Caso 3, Caso 3, Caso 1

(E) Caso 1, Caso 3, Caso 3, Caso 2

---

**Questão 9.** Guilherme está aprendendo sobre as estruturas de repetição “repita” e “enquanto”, por isso ele fez dois algoritmos para treinar os seus usos:

```
01. Programa1
02.   ler a, b
03.   copia_b <- b
04.   se a > b
05.     b <- a
06.     a <- copia_b
07.   fim_se
08.   enquanto b > 0
09.     b <- b - a
10.   fim_enquanto
11.   retorne b
12. fim_programa1
```

```
01. Programa2
02.   ler a, b
03.   copia_b <- b
04.   se a > b
05.     b <- a
06.     a <- copia_b
07.   fim_se
08.   repita
09.     b <- b - a
10.   até b < 0
```



## 9 | XI Olimpíada Cearense de Informática

11. fim\_repita
12. retorne b
13. fim\_programa2

Ao rodar os dois programas com as entradas sendo  $a = 25$  e  $b = 2$ , Guilherme gostaria de saber se um dos programas é finalizado com uma menor quantidade de iterações e além disso, quais serão os respectivos retornos do Programa 1 e do Programa 2.

- (A) Programa 1; -1 e 1.  
(B) Programa 2; 1 e 1.  
(C) Programa 1; 1 e -1.  
(D) Ambos programas rodam com a mesma quantidade de iterações; -1 e 1.  
(E) Ambos programas rodam com a mesma quantidade de iterações; -1 e -1.
- 

**Questão 10.** Você está resolvendo um problema de programação em que precisa criar um programa para gerar a sequência de Fibonacci. A sequência de Fibonacci é definida da seguinte forma:

- Os dois primeiros números são 0 e 1;
- Os números subseqüentes são a soma dos dois números anteriores.

Você decide implementar o programa com uma função recursiva.

A função recebe um número inteiro  $n$  como entrada e retorna o  $n$ -ésimo número da sequência de

Fibonacci. No entanto, você percebe que a função está muito lenta quando  $n$  é grande.

Qual é o

principal motivo para a lentidão da função recursiva ao calcular números grandes na

sequência de

Fibonacci?

- (A) Má implementação da função recursiva.  
(B) Excesso de uso de memória.  
(C) Complexidade de tempo exponencial.  
(D) Falta de otimização do compilador.  
(E) Problemas de memória cache.
- 

**Questão 11.** Matriz é uma tabela organizada em linhas e colunas no formato  $m \times n$ , onde  $m$  representa o número de linhas (horizontal) e  $n$  o número de colunas (vertical). A função

## 10 | XI Olimpíada Cearense de Informática

das matrizes é relacionar dados numéricos. Por isso, o conceito de matriz não é só importante na Matemática, mas também em outras áreas, já que as matrizes têm diversas aplicações, por exemplo, na programação é possível manipular matrizes e aplicar operações entre elas para chegar aos resultados que se deseja. Analise a função abaixo e responda:

**Função Faz\_algo(matrizA, matrizB):**

**Se** TamColunas(matrizA) != TamLinhas(matrizB):

**Retornar** "Erro!"

**Fim Se**

linhasNovaMatriz = TamLinhas(matrizA)

colunasNovaMatriz = TamColunas(matrizB)

matrizNova = MatrizVazia(linhasNovaMatriz, colunasNovaMatriz)

**Para** i de 1 até linhasNovaMatriz **faça**:

**Para** j de 1 até colunasNovaMatriz **faça**:

**Para** k de 1 até TamColunas(matrizA) **faça**:

matrizNova[i][j] += matrizA[i][k] \* matrizB[k][j]

**Fim Para**

**Fim Para**

**Fim Para**

**Retornar** matrizNova

**Fim Função**

Sabendo-se que:

matrizA = [[0,2,3],[1,4,4],[1,5,1]]

matrizB = [[4,2,0],[2,1,3],[0,2,4]]

Qual será a matriz retornada pela função Faz\_algo(matrizA, matrizB)?

a)

$$\begin{bmatrix} 4 & 4 & 3 \\ 3 & 5 & 7 \\ 1 & 7 & 5 \end{bmatrix}$$

b)

$$\begin{bmatrix} 4 & 8 & 18 \\ 12 & 14 & 28 \\ 14 & 9 & 19 \end{bmatrix}$$

c)

$$\begin{bmatrix} 2 & 16 & 20 \\ 4 & 23 & 13 \\ 6 & 28 & 12 \end{bmatrix}$$

d)

$$\begin{bmatrix} 4 & 4 & 3 \\ 3 & 5 & 7 \\ 1 & 7 & 5 \end{bmatrix}$$

e)

$$\begin{bmatrix} 4 & 8 & 18 \\ 12 & 16 & 20 \\ 14 & 9 & 19 \end{bmatrix}$$

**Questão 12.** Analise o algoritmo abaixo:

1. **Alg: FazAlgo (vetor A)**
2.  $n \leftarrow \text{comprimento}[A]$
3. **Para**  $i \leftarrow 1$  até  $n-1$  **faça**
4.     menor  $\leftarrow i$
5.     **Para**  $j \leftarrow i+1$  até  $n$  **faça**
6.         **Se**  $A[j] < A[\text{menor}]$  **então**
7.             menor  $\leftarrow j$
8.     **Fim Se**
9.     **Fim Para**
10.    Temp  $\leftarrow A[i]$
11.     $A[i] \leftarrow A[\text{menor}]$
12.     $A[\text{menor}] \leftarrow \text{Temp}$
13.    **Fim Para**

De acordo com as estruturas de repetição presentes no código supracitado, qual dos itens abaixo possui a conclusão correta sobre a corretude do algoritmo?

- (A) No final de cada iteração do laço externo “para”, o subvetor  $A[1, \dots, i-1]$  consiste nos  $j-1$  elementos menores do vetor  $A[1, \dots, n]$  e este subvetor está ordenado decrescentemente.
- (B) Após os primeiros  $n-1$  elementos, o subvetor  $A[1, \dots, n-1]$  contém os  $n-1$  elementos menores ordenados, e o elemento  $A[n]$  tem de ser o menor de todos.
- (C) No laço interno “para”, nenhum elemento do vetor  $A[i, j+1]$  é menor que  $A[\text{menor}]$ .
- (D) No início de cada iteração do laço interno “para”, o subvetor  $A[1, \dots, i-1]$  consiste nos  $j-1$  elementos menores do vetor  $A[1, \dots, n]$  e este subvetor está ordenado.
- (E) Nenhuma das alternativas anteriores.
- 

**Questão 13.** Observe o seguinte algoritmo e, dadas as entradas disponibilizadas, informe qual a saída correta que deverá ser exibida ao final da execução:

1.  $A \leftarrow [1, 2, 3, 4, 5, 6, 7, 8, 9]$
2.  $B \leftarrow 137$
3.  $C \leftarrow 22$
4.  $\text{result} \leftarrow 1$
- 5.
6. **enquanto** não\_vazia(A):
7.      $\text{result} \leftarrow \text{result} * A.\text{retirar\_primeiro}()$
- 8.

## 12 | XI Olimpíada Cearense de Informática

9. `result <- result // B`
10. `result <- result + C`
- 11.
12. `imprima (result)`

### Descrição:

- ' <- ' = Atribui um valor a uma variável
- ' \* ' = Retorna a multiplicação de dois números
- ' // ' = Retorna a divisão inteira de dois números
- ' + ' = Retorna a soma de dois números
- ' não\_vazia() ' = Retorna verdadeiro se a lista passada como argumento tiver algum conteúdo
- ' retirar\_primeiro() ' = Retira o primeiro item da lista e o retorna

- (A) 2480
  - (B) 2670
  - (C) 3720
  - (D) 3960
  - (E) 4130
- 

**Questão 14.** Um professor de informática decidiu lançar um desafio a fim de testar os conhecimentos dos alunos. Criou uma linguagem onde um código binário fornece uma palavra qualquer. Sabendo que nessa linguagem:

- 1001100011001110001 corresponde a palavra CABIDE.

A soma dos números 1 presentes no código da palavra CHEFE é:

- ( A ) 5;
  - ( B ) 6;
  - ( C ) 7;
  - ( D ) 8;
  - ( E ) 9.
- 

**Questão 15.** Na Ciência da Computação, mais especificamente nas Ciências de Construção de e Análise de Algoritmos, existe uma classe de soluções denominadas "Algoritmos de Busca". Tais soluções vislumbram técnicas para encontrar a posição específica de um determinado elemento em um Vetor. Dentre esses algoritmos, um bastante simples, no entanto prático, trata-se da "Busca Binária". Sua abordagem consiste em verificar em dividir o Vetor em sub-regiões, limitadas por dois índices, e verificar se o

### 13 | XI Olimpíada Cearense de Informática

elemento está no centro dessa sub-região, se não, uma sub-região menor é limitada. Saiba que essa solução funciona apenas para vetores previamente ordenados, pois então é possível prever como as sub-regiões são limitadas, e que o vetor começa pelo índice de número 1 e termina no índice de número  $n$ .

**Algoritmo:** Busca\_Binária.

**Entrada:** um vetor de números “V”, de tamanho “n”, e um valor “K”.

**Saída:** um índice “J”, onde o elemento “K” se encontra em “V”.

INTEIRO esq = 1, dir = n, mid

**ENQUANTO** esq <= dir **FAÇA**

    mid esq + (dir – esq) / 2;

**SE** K = V[mid] **ENTÃO**

**RETORNE** mid

**FIM SE**

**SE** K < V[mid] **ENTÃO**

        dir mid – 1

**SENAO**

        esq mid + 1

**FIM SENAO**

**FIM\_ENQUANTO**

**RETORNE** -1 // Valor de Erro: O elemento “K” não está no veto

Imagine que alguém decida executar esse algoritmo e, como entrada, ele envia um vetor  $V = \{3, 6, 7, 9, 10, 11, 16, 24, 33, 41, 50\}$  e deseja saber a posição do elemento  $K = 24$ . Assinale a alternativa que contém quantidade de valores que “mid” recebeu antes de encontrar “K” e os valores finais das variáveis “esq” e “dir”:

- ( A ) mid recebeu 4 atribuições; esq = 7 e dir = 9;
- ( B ) mid recebeu 6 atribuições; esq = 10 e dir = 8;
- ( C ) mid recebeu 6 atribuições; esq = 6 e dir = 10;
- ( D ) mid recebeu 4 atribuições; esq = 8 e dir = 8;
- ( E ) mid recebeu 2 atribuições; esq = 2 e dir = 9.

---

**Questão 16.** Na figura abaixo, as letras foram dispostas no formato de um triângulo, seguindo uma determinada restrição.

14 | XI Olimpíada Cearense de Informática

$$\begin{array}{c} ? \\ C + P \\ H + D + F \\ B + J + A + E \end{array}$$

Considerando que foi utilizada uma ordem alfabética onde as letras “K”, “W” e “Y” são excluídas, determine qual letra é adequada para substituir o ponto de interrogação:

- ( A ) G;
  - ( B ) X;
  - ( C ) M;
  - ( D ) S;
  - ( E ) Q.
- 

**Questão 17.** Um aluno têm seis disciplinas na sua escola, sabendo que suas notas têm as seguintes condições:

- Disciplina 1 : Nota 4 ou 4,5;
- Disciplina 2: Nota 5 ou 5,5;
- Disciplina 3: Nota 6 ou 6,5;
- Disciplina 4: Nota 7 ou 7,5;
- Disciplina 5: Nota 8 ou 8,5;
- Disciplina 6: Nota 9 ou 9,5.

Quantas possíveis combinações de notas esse aluno pode ter sabendo que a média total das disciplinas é menor ou igual a 6,84 ?

- ( A ) 21;
  - ( B ) 32;
  - ( C ) 42;
  - ( D ) 57;
  - ( E ) 58.
- 

**Questão 18.** Laryssa é a melhor aluna da turma da professora Ada, uma excelente professora de matemática. Por ter esse mérito, Laryssa foi convidada pela sua professora para testar um código que a docente criou para auxiliar a turma no processo para

## 15 | XI Olimpíada Cearense de Informática

aprender sobre matrizes. Dito isso, observe o código da professora Ada, que se encontra logo abaixo:

**Programa:**

```
01.
02. Procedimento: matriz(x: inteiro)
03.
04.   para i ← 0 até x-1 faça
05.     para j ← 0 até x-1 faça
06.       se (i = j) então
07.         imprima (j);
08.       senão
09.         se (i = 0) ou (i = x-1) então
10.           imprima (“&”);
11.         senão
12.           se (j = 0) ou (i = x-1) então
13.             imprima("&");
14.           senão
15.             imprima(" ")
16.           fim_se
17.         fim_se
18.       fim_se
19.     fim_para
20.   fim_para
21. fim_procedimento
22.
23. Principal
24. imprima (“Entre um valor inteiro y: “)
25. leia (y)
26. matriz(y)
27. Fim_principal
```

Ao estudar o procedimento, o que Laryssa pôde concluir sobre o programa?

- (A) Ao rodar o programa, Laryssa percebeu que trata-se um código que destaca, com números em ordem decrescente de  $x-1$  a 0, a diagonal principal de uma matriz.
- (B) Ao rodar o programa, Laryssa percebeu que houve uma passagem por referência entre o programa principal e o procedimento.
- (C) Laryssa observou que o programa, além de apresentar uma passagem por parâmetros, imprime o seguinte padrão quando o usuário entrar o número 5:

## 16 | XI Olimpíada Cearense de Informática

```
0 & & & &  
& 1 & & &  
& & 2 & &  
& & & 3 &  
& & & & 4
```

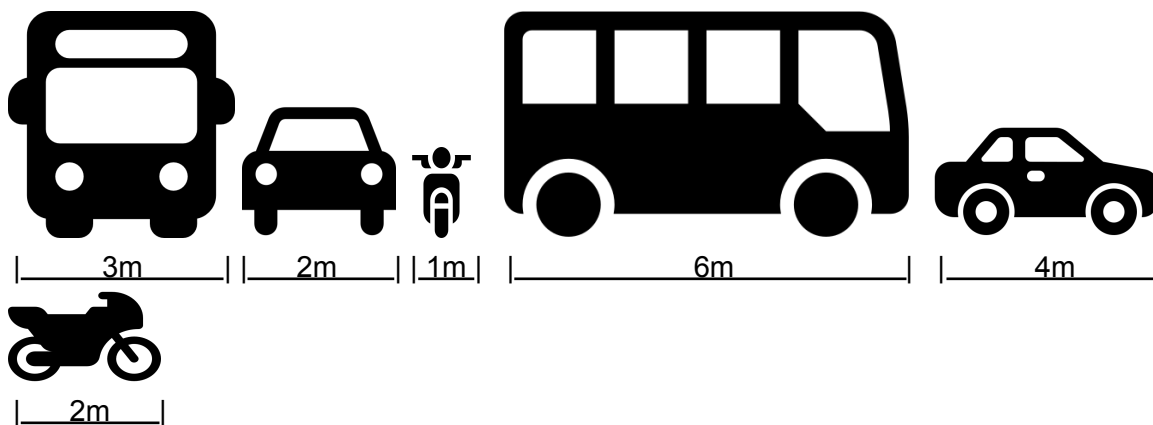
(D) Laryssa concluiu que o programa apresenta um erro na linha vinte e cinco, pois o procedimento recebe x e não y.

(E) Laryssa observou que o programa, além de destacar a diagonal principal com números em ordem crescente, imprime o seguinte padrão quando o usuário entrar o número 5:

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

**Questão 19.** Alocar veículos em uma rua é uma questão relevante, afinal, se pudermos alocá-los de maneira eficiente seremos capazes de contribuir para a mobilidade urbana.

Considere que uma motocicleta ocupa 1 (um) metro de comprimento e 2 (dois) metros de largura, um carro 4 (quatro) metros de comprimento e 2 (dois) metros de largura e um ônibus 6 (seis) metros de comprimento e 3 (três) metros de largura.



Um ônibus comporta até 38 pessoas, um carro comporta até 5 (cinco) pessoas e uma motocicleta comporta até 2 (duas) pessoas.

Em uma via de 12 metros de largura e 20 metros de comprimento, é possível dispor 360 pessoas em veículos, de modo que a quantidade de carros, motos e ônibus seja igual? Se sim, haverá sobra de espaço na via?



## 17 | XI Olimpíada Cearense de Informática

- (A) 24 veículos e pode haver sobra de espaço.
  - (B) 24 veículos e não haverá sobra de espaço.
  - (C) 36 veículos e pode haver sobra de espaço.
  - (D) 36 veículos e não haverá sobra de espaço.
  - (E) Não é possível, não há espaço suficiente na via para comportar os veículos dessa maneira.
- 

**Questão 20.** Considere uma *string* como um vetor de caracteres, exemplo:

String var1					
caracteres	C	E	A	R	A
índice	0	1	2	3	4

De modo que se a variável acima se chame **var1**, **var1[0] = 'C'** e **var1 = ['C', 'E', 'A', 'R', 'A']**.

Considere também as seguintes funções relacionadas a *strings*:

1. **var1.tam() = 5**
2. **String var2 ← stringInic(5), var2 = [",", ",", ",", ",", ",", "]**

Sabendo disso, faça a análise da execução do seguinte programa:

```
Funcao(String s, String i, Inteiro n, Inteiro p) {
    SE (p = 0) FAÇA {
        i[n - p] = s[p];
    }
    SENAO {
        Funcao(s, i, n, p-1);
        i[n - p] = s[p];
    }
}
Programa(String nome) {
    Inteiro n ← nome.tam();
    String x ← stringInic(n);
    Funcao(nome, x, n, n);
    RETORNA ( x );
}
```

Assim, podemos concluir que:

**18 | XI Olimpíada Cearense de Informática**

- (A) Funciona perfeitamente e retorna um vetor de inteiros de tamanho correspondente ao de “nome”.
- (B) Não funciona corretamente e retorna o inverso da *string* “nome”.
- (C) Funciona e retorna o inverso da *string* “x”.
- (D) Se a *string* “nome” fosse ['O', 'C', 'I'], então o retorno seria ['i', 'c', 'o'].
- (E) Não funciona e retorna um erro.