



UNIVERSIDADE ANHANGUERA-UNIDERP
UNIDADE MATRIZ

Avenida Ceará nº 333 – Bairro Miguel Couto
Campo Grande - MS, CEP 79003-010, Telefone: (67) 3348-8000

PROVA N1

ENGENHARIA DA COMPUTAÇÃO			NOTA:
Acadêmico(a):			
CPF:			
Disciplina: ALGORITMOS E LINGUAGEM DE PROGRAMAÇÃO			
Semestre:	Turno:	Data:	

REGULAMENTO DA PROVA:

- Identifique a sua prova preenchendo os campos em brancos indicados neste cabeçalho (Acadêmico, CPF, Semestre, Turno e Data);
- Esta prova não é de consulta;
- Não haverá esclarecimento quanto à interpretação da pergunta;
- Em cima da carteira, apenas lápis, borracha e caneta;
- O celular deve permanecer desligado e dentro da bolsa/mochila sob pena de ser aplicada nota mínima;
- O valor de cada questão está colocado ao lado da mesma;
- Leia atentamente o enunciado de cada questão, procure refletir antes de responder e busque atender exatamente ao solicitado;
- Todas as questões da prova (objetivas, descritivas ou que envolvam cálculos) devem ser respondidas no gabarito à caneta, sem rasuras e sem o uso de corretivo, pois ficará sujeito a ser cancelada pelo(a) professor(a);
- Caso uma questão seja cancelada ou anulada, a nota será distribuída pelas outras questões da prova, alterando e elevando o peso das questões restantes.
- A permanência mínima dos alunos em sala durante a prova é de 50 minutos contados a partir do início da prova, sendo proibida a saída dos mesmos antes desse tempo;
- É obrigatória a presença do aluno no período de revisão de nota/prova. Caso o aluno não compareça à revisão de nota/prova, não terá direito à revisão da prova posteriormente, sendo permitido somente a revisão de nota solicitada através do Portal do Aluno, 48 horas após o lançamento da nota no portal.
- A devolução da prova ocorrerá na data de revisão de prova e será feita pelo professor da disciplina. Terão acesso a prova somente os alunos devidamente matriculados.

GABARITO

- | | |
|-------------------------|-------------------------|
| 01. (A) (B) (C) (D) (E) | 06. (A) (B) (C) (D) (E) |
| 02. (A) (B) (C) (D) (E) | 07. (A) (B) (C) (D) (E) |
| 03. (A) (B) (C) (D) (E) | 08. (A) (B) (C) (D) (E) |
| 04. (A) (B) (C) (D) (E) | 09. (A) (B) (C) (D) (E) |
| 05. (A) (B) (C) (D) (E) | 10. (A) (B) (C) (D) (E) |

ACERTOS: _____

ACERTOS: _____



UNIVERSIDADE ANHANGUERA-UNIDERP
UNIDADE MATRIZ

Avenida Ceará nº 333 – Bairro Miguel Couto
Campo Grande - MS, CEP 79003-010, Telefone: (67) 3348-8000

PROVA N1

QUESTÃO 01

Analise as afirmativas abaixo.

- (I) Um algoritmo não representa necessariamente um programa de computador, e sim os passos para realizar determinada tarefa.
- (II) Um algoritmo determina o valor necessário para elaborar uma tarefa.
- (III) A implementação para um mesmo algoritmo pode ser feita em várias linguagens de programação diferentes.
- (IV) Diferentes algoritmos podem realizar a mesma tarefa usando um conjunto diferenciado de instruções.

Assinale a alternativa que apresenta a sequência correta de I a IV.

- | | | | | |
|----|---|---|---|------|
| A) | F | V | F | F |
| B) | V | F | F | V |
| C) | F | V | V | F |
| D) | V | V | V | V |
| E) | V | F | V | V OK |

QUESTÃO 02

Sobre algoritmos, analise as afirmativas abaixo.

- (I) São procedimentos precisos, não ambíguos, eficientes e corretos.
- (II) Um algoritmo representa os passos necessários para realização de uma tarefa.
- (III) São representados somente por fluxogramas.

Assinale a alternativa que apresenta a sequência correta de cima para baixo.

- | | | | |
|----|---|---|------|
| A) | F | V | V |
| B) | F | F | V |
| C) | V | F | F |
| D) | V | V | F OK |
| E) | V | F | V |

QUESTÃO 03

Sobre os algoritmos de computadores, marque a alternativa correta.

- A) Algoritmos de computador são utilizados exclusivamente para resolver problemas da computação, como por exemplo a detecção de códigos maliciosos em computadores.
- B) Nos algoritmos, as variáveis são, notadamente, utilizadas para representar valores previamente conhecidos.
- C) Um algoritmo de computador é composto por várias etapas que, em conjunto, executam uma determinada tarefa. OK
- D) Algoritmos de computador não podem ser definidos como um conjunto de instruções que o computador terá que executar para resolver um problema.
- E) N.D.A.

QUESTÃO 04

Considere o programa na linguagem C:

```
#include <stdio.h>
int main(){
    int a, b, c, d;
    printf("Digite três números inteiros: ");
    scanf("%d %d %d", &a, &b, &c);
    printf("\n");
    a = b; d = c; d = a; c = b; b = d; b = a;
    a = c; d = b; c = d; c = a; b = c; a = d;
    printf("%d %d %d %d \n", a, b, c, d);
    return 0;
}
```

Executando e digitando como entrada os números 5, 31, 12, a saída será:

- | | | | | |
|----|----|----|----|-------|
| A) | 5 | 12 | 31 | 5 |
| B) | 12 | 31 | 5 | 12 |
| C) | 31 | 31 | 31 | 31 OK |
| D) | 5 | 12 | 5 | 12 |
| E) | 12 | 12 | 12 | 12 |

PROVA N1

QUESTÃO 05

Em relação à programas escritos na Linguagem C/C++, considere:

- I. Toda repetição pode ser representada por uma estrutura do tipo while ou do-while, sendo que a primeira repete somente quando a condição é falsa e a segunda somente quando a condição é verdadeira.
- II. Na estrutura de repetição while o bloco de repetição pode ser executado várias vezes ou até nenhuma vez. A condição é testada antes de entrar na estrutura de repetição.
- III. A estrutura de repetição do-while efetua um teste lógico no fim do laço, garantindo que pelo menos uma vez as instruções deste serão executadas.
- IV. Para se impedir a situação conhecida como loop infinito é necessário que, nos laços condicionais, a variável que é testada esteja sempre associada a uma instrução que a atualize no interior do laço.

É correto o que se afirma APENAS em:

- A) I, II e IV.
- B) I, II e III.
- C) II e III.
- D) II, III e IV. OK
- E) Todas as afirmações são verdadeiras.

QUESTÃO 06

A estrutura condicional possibilita a escolha de um grupo de ações e estruturas a serem executadas quando determinadas condições são ou não satisfeitas. A estrutura condicional pode ser simples ou composta. Nesse contexto, analise as asserções a seguir e marque (V) verdadeiro ou (F) falso:

- I - A estrutura condicional simples executa um comando ou vários comandos se a condição for verdadeira.
- II - Se a condição for falsa, a estrutura condicional simples é finalizada após executar todos os comandos.
- III - O comando que define a estrutura condicional simples é representado pela palavra IF.
- IV - A estrutura condicional composta segue o mesmo princípio da estrutura condicional simples, com a diferença de que quando a condição não é satisfeita, o mesmo comando é executado até a condição ser satisfeita.
- V - O comando que define a estrutura condicional composta é representado pelas palavras IF e ELSE.

Assinale a alternativa que contém a sequência correta:

- A) F-V-F-V-F.
- B) V-V-F-F-V.
- C) F-F-V-V-F.
- D) V-F-F-F-V.
- E) V-F-V-F-V. OK

PROVA N1

QUESTÃO 07

A maior parte das pessoas não ligadas à área de informática ignoram o potencial dos computadores e imagina que eles são capazes de tratar apenas com dados numéricos. As informações tratadas em um algoritmo podem ser classificadas em dois tipos: Variáveis e Constantes. Acerca das variáveis, assinale a alternativa incorreta.

- A) O conteúdo de uma variável deverá ser um dado do tipo usado na sua criação.
B) O conteúdo de uma variável é substituído por outro conteúdo que lhe venha a ser atribuído.
C) É possível nomear uma variável sem qualquer restrição sintática. **OK**
D) O uso do nome de uma variável em uma expressão significa o uso do seu conteúdo (naquele momento) dentro da expressão.
E) O uso de um conteúdo de variável em uma expressão não modifica o seu valor.

QUESTÃO 08

Você precisa implementar um programa em C que utilize a instrução while para somar todos os números inteiros de 1 a 10, inclusive. Qual das opções a seguir implementa essa lógica corretamente?

<p>A)</p> <pre>int i = 1, soma = 0; while (i < 10) { soma += i; i++; } printf("%d", soma);</pre>	<p>B)</p> <pre>int i = 0, soma = 0; while (i <= 10) { soma += i; i++; } printf("%d", soma);</pre>	<p>C)</p> <pre>int i = 1, soma = 0; while (i <= 10) { soma += i; i++; } printf("%d", soma);</pre>
<p>D)</p> <pre>int i = 1, soma = 0; while (i <= 10) { i++; soma += i; } printf("%d", soma);</pre>	<p>E)</p> <pre>int i = 1, soma = 0; while (i < 10) { soma += i; i += 2; } printf("%d", soma);</pre>	

GABARITO

Letra C, pois em comparação com a letra B, tem uma iteração a menos, ou seja, seria mais eficiente.

<p>Testando a Letra A -</p> <pre>valores - i[1] - soma[0] valores - i[2] - soma[1] valores - i[3] - soma[3] valores - i[4] - soma[6] valores - i[5] - soma[10] valores - i[6] - soma[15] valores - i[7] - soma[21] valores - i[8] - soma[28] valores - i[9] - soma[36] 45</pre>	<p>Testando a Letra B -</p> <pre>valores - i[0] - soma[0] valores - i[1] - soma[0] valores - i[2] - soma[1] valores - i[3] - soma[3] valores - i[4] - soma[6] valores - i[5] - soma[10] valores - i[6] - soma[15] valores - i[7] - soma[21] valores - i[8] - soma[28] valores - i[9] - soma[36] valores - i[10] - soma[45] 55</pre>	<p>Testando a Letra C -</p> <pre>valores - i[1] - soma[0] valores - i[2] - soma[1] valores - i[3] - soma[3] valores - i[4] - soma[6] valores - i[5] - soma[10] valores - i[6] - soma[15] valores - i[7] - soma[21] valores - i[8] - soma[28] valores - i[9] - soma[36] valores - i[10] - soma[45] 55</pre>
<p>Testando a Letra D -</p> <pre>valores - i[1] - soma[0] valores - i[2] - soma[2] valores - i[3] - soma[5] valores - i[4] - soma[9] valores - i[5] - soma[14] valores - i[6] - soma[20] valores - i[7] - soma[27] valores - i[8] - soma[35] valores - i[9] - soma[44] valores - i[10] - soma[54] 65</pre>	<p>Testando a Letra E -</p> <pre>valores - i[1] - soma[0] valores - i[3] - soma[1] valores - i[5] - soma[4] valores - i[7] - soma[9] valores - i[9] - soma[16] 25</pre>	



UNIVERSIDADE ANHANGUERA-UNIDERP
UNIDADE MATRIZ

Avenida Ceará nº 333 – Bairro Miguel Couto
Campo Grande - MS, CEP 79003-010, Telefone: (67) 3348-8000

PROVA N1

QUESTÃO 09

Observe o código a seguir, que recebe um número positivo e imprime seus divisores. Exemplo: os divisores do número 66 são: 1, 2, 3, 6, 11, 22, 33 e 66.

```
#include <stdio.h>
int main(){
    int n, i;
    printf("Digite um número inteiro: ");
    scanf("%d", &n);
    for(i = 1; i <= n; i++){
        if(n % i == 0)
            printf("%d ", i);
    }
    return 0;
}
```

Executando e digitando como entrada o número 324, os 7 primeiros caracteres da saída serão:

- | | | | | | | | | |
|----|---|---|----|----|----|----|-----|----|
| A) | 1 | 3 | 5 | 7 | 9 | 12 | 35 | |
| B) | 1 | 2 | 4 | 6 | 12 | 36 | 72 | |
| C) | 1 | 2 | 7 | 14 | 23 | 46 | 161 | |
| D) | 1 | 2 | 3 | 4 | 6 | 9 | 12 | OK |
| E) | 1 | 4 | 12 | 24 | 32 | 72 | 128 | |

PROVA N1

QUESTÃO 10 – ANULADA

Em um programa de C, considere que você deseja verificar a idade de uma pessoa para determinar se ela ainda é criança, adolescente, adulta ou idosa. As idades são classificadas da seguinte maneira:

- Criança: 0 a 12 anos
- Adolescente: 13 a 17 anos
- Adulto: 18 a 59 anos
- Idoso: 60 anos ou mais

Qual das opções a seguir representa corretamente o uso das instruções if, else if e else para implementar essa lógica em C?

<p>A)</p> <pre>if (idade < 13) { printf("Criança"); } else if (idade < 18) { printf("Adolescente"); } else if (idade < 60) { printf("Adulto"); } else { printf("Idoso"); }</pre>	<p>D)</p> <pre>if (idade >= 0 && idade <= 12) { printf("Criança"); } else if (idade >= 13 && idade <= 17) { printf("Adolescente"); } else if (idade >= 18 && idade <= 59) { printf("Adulto"); } else if (idade >= 60) { printf("Idoso"); }</pre>
<p>B)</p> <pre>if (idade <= 12) { printf("Criança"); } else if (idade <= 17) { printf("Adolescente"); } else if (idade <= 59) { printf("Adulto"); } else if (idade >= 60) { printf("Idoso"); }</pre>	<p>E)</p> <pre>if (idade <= 12) { printf("Criança"); } else { printf("Idoso"); }</pre>
<p>C)</p> <pre>if (idade >= 60) { printf("Idoso"); } else if (idade >= 18) { printf("Adulto"); } else if (idade >= 13) { printf("Adolescente"); } else { printf("Criança"); }</pre>	

Problema: possui mais de uma questão correta, considerando apenas a lógica apresentada, o que pode induzir ao erro. Letras A, B, C, D.