Disciplina:	DEC7531 Linguagem de Programação I	**Data:**	01/10/2020
Nome:	Matheus Avila Naspolini	**Matrícula:**	19207851

PROVA 1

Obs.: As questões teóricas devem ser respondidas no arquivo de texto e enviadas pelo Moodle no formato .**doc**, **.odt** ou **.pdf**.

As questões praticas devem ser enviadas pelo Moodle no formato **.c**

Questão 1 (teórica, 1 ponto):

Como funciona mecanismo de funções em linguagem C e em quais situações seu uso é indicado?

Uma função é um modulo (sequência de código) criado para resolver uma tarefa simples e bem definida. Os operadores que fazem parte do corpo da função são escritos em um único lugar e uma única vez e permanecem ocultos para resto do código. As funções são “chamadas” ou “ativadas” ao longo da execução do programa, no momento da chamada é especificado: o nome da função e os argumentos (a informação que é necessária para que a função cumpra seu proposito). Depois da execução, a função retorna o resultado para o ponto em qual foi chamada. O seu uso é indicado para desenvolver e manter um programa grande, pois é mais fácil construí-lo a partir de pequenas partes ou componentes, sendo cada uma delas mais fácil de manipular que o programa original. Essas pequenas partes são as funções.

Questão 2 (teórica, 1 ponto):

Dado um protótipo da função

```
my_fun (double x, double y, int z);
```

Qual das afirmações esta correta:

a) função espera receber três variáveis do tipo double e não vai retornar nenhum valor

b) função espera receber duas variáveis do tipo double, uma variável do tipo int e vai retornar um valor do tipo double

c) função espera receber duas variáveis do tipo double, uma variável do tipo int e vai retornar um valor do tipo int

d) função espera receber duas variáveis do tipo double, uma variável do tipo int e não vai retornar nenhum valor

Resposta:

Letra C, pois quando não especificado, uma função sempre retorna uma variável do tipo int.

Questão 3 (teórica, 1 ponto):

É possível fazer com que uma variável declarada dentro de uma função preserva o seu valor ao longo da execução do programa inteiro e não somente durante a execução da função onde essa variável foi declarada?

Justifique a reposta.

Sim, usando a classe de alocação static. Essa classe indica para o compilador que a variável local deve existir durante a execução de programa inteiro. Exemplo: “static int i”, uma variável desse tipo vai preservar seu valor entre as chamadas de função.

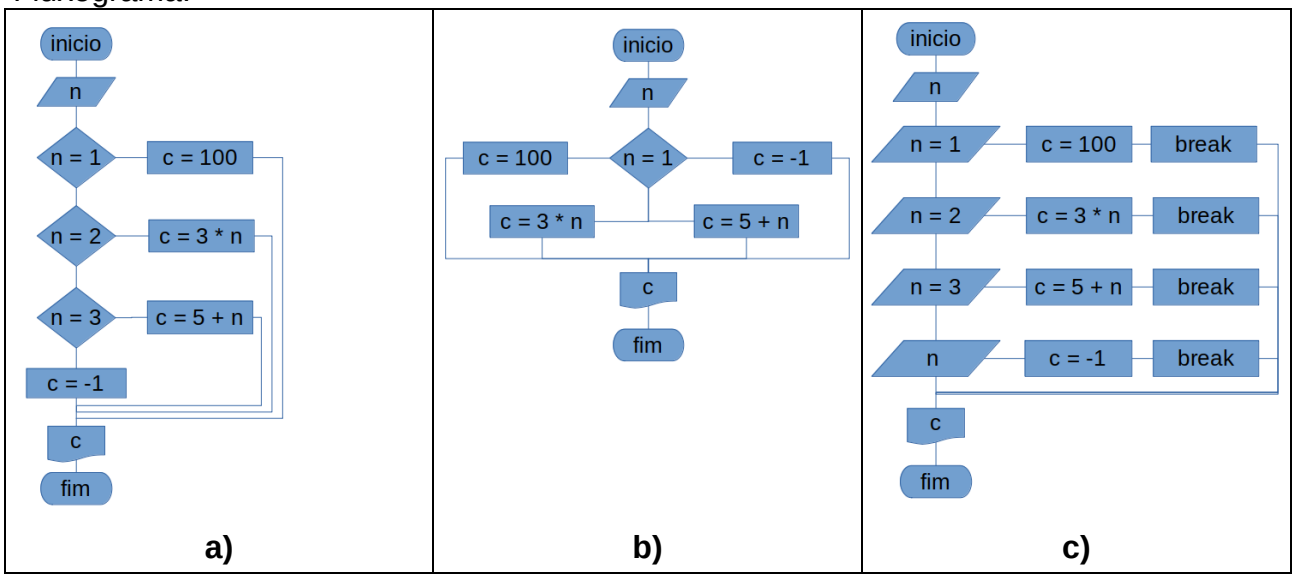
Questão 4 (teórica, 1 ponto):

Analise o código e escolha qual dos fluxogramas representa ele de forma correta. Justifique a escolha.

Código:

```c
#include <stdio.h>

int main()
{
    int n, c;

    printf("Digite n ");
    scanf("%i", &n);

    switch(n)
    {
      case 1:
          c = 100;
          break;
      case 2:
          c = 3 * n;
          break;
      case 3:
          c = 5 + n;
          break;
      default:
          c = -1;
          break;
    }

    printf("c = %i", c);

    return 0;
}
```

Fluxograma:



Resposta:

Letra A, pois os símbolos das situações de entrada de dados, saída de dados, tomadas de decisão e as próprias linhas estão corretas. Na letra B por exemplo, o fluxograma não representa as tomadas de decisão do caso switch. Na Letra C, ele não representa as tomadas de decisão e coloca desnecessariamente o comando “break” no fluxograma.

Questão 5 (pratica, 3 pontos):

Criar um vetor com 10 elementos e preencher ele com números inteiros definidos pelo usuário.

a) Calcular a soma dos elementos que são **menores** do que o elemento anterior.

Exemplo:

1	7	3	2	4	1	7

Soma = 3 + 2 + 1

b) Substituir todos os elementos menores do que um número **k** definido pelo usuário pelo **menor elemento** do vetor utilizando **uma função** que vai receber o vetor e vai retornar o valor do menor elemento.

Exemplo:

11	6	7	10	-4	9	6

Resultado se **k = 8**:

11	-4	-4	10	-4	9	-4

Questão 6 (pratica, 3 pontos):

Criar uma matriz **M 3x3** preencher ela com números inteiros definidos pelo usuário.

a) Calcular a quantidade de elementos ímpares em cada coluna

Exemplo:

1	2	3
5	3	6
4	1	2

Resultado:
Coluna 0 – 2 elemento(s)
Coluna 1 – 2 elemento(s)
Coluna 2 – 1 elemento(s)

b) Trocar os elementos da linha 0 e linha 1 de matriz de lugar.

Exemplo:

1	2	3
4	5	6
7	8	9

Resultado:

7	8	9
4	5	6
1	2	3