Ministério da Educação
Universidade do Estado de Santa Catarina
Centro de Educação do Planalto Norte - CEPLAN

Nome:	**Data:** 14/05/2018
Matrícula:	**Nota da Avaliação:**
Disciplina: Nome da Disciplina	
Professor: Nome do Professor	*Rúbrica do Professor*

`Orientações Gerais:`

1 - **DESLIGUE E GUARDE O CELULAR**.

2 - Preencha seu nome e número de Registro Acadêmico nas folhas utilizadas.

3 - A interpretação das questões faz parte do processo de avaliação, não sendo permitidas consultas ou comunicação entre alunos.

4 - Os exercícios devem sem apresentados seguindo a estrutura da linguagem vista em sala de aula.

5 - Qualquer dúvida o aluno deverá chamar o(a) professor(a).

Questão:	1	2	3	4	5	6	Total
Pontos:	10	5	5	10	40	30	100
Nota:							

1. (10 Pontos) Considerando o código abaixo, e os conceitos apresentados em aula, apresente o teste de mesa que ilustre o comportamento do programa na memória. Considere como valores de entrada o conjunto de 5 elementos: $\{4, 3, 5, 2, 1\}$.

```c
#include <stdio.h>
#include <stdlib.h>
void lervetor(int *v, int tam){
    int i;
    for (i=0; i<tam; i++){
        scanf("%d", &v[i]);
    }
}
void maiormenor(int *v, int tam, int* maior, int* menor){
    int i;
    *maior = v[0];
    *menor = v[0];
    for (i=1;i<tam;i++){
        if (v[i] > *maior){
            *maior = v[i];
        }
        if (v[i] < *menor){
            *menor = v[i];
        }
    }
}
int main (){
    int n;
    int *vet = NULL;
    int maior;
    int menor;
    printf("Informe a quantidade de elementos: ");
    scanf("%d", &n);
    vet = (int*) malloc (n * sizeof(int));
    lervetor(vet, n);
    maiormenor(vet, n, &maior, &menor);
    printf("O maior valor informado foi: %d\n", maior);
    printf("O menor valor informado foi: %d\n", menor);
    return 0;
}
```

2. (5 Pontos) Escreva as instruções (em linguagem C) para realizar as seguintes tarefas:

    a) Declare um vetor de inteiros de 100 elementos;

b) Imprima na tela o valor da sétima posição do vetor;

c) Atribua o valor 10 na quarta posição do vetor;

d) Multiplique os valores das posições de indice 5 e 6 e armazene o resultado na posição de indice 8;

e) Troque os valores da primeira e última posição;

f) Some todos os valores do vetor;

3. (5 Pontos) Descreva o funcionamento das funções `malloc` e `free` e apresente exemplos de sua utilização.

4. Tipos estruturados:
   (a) (2 Pontos) Descreva as vantagens da utilização de tipos estruturados.
   (b) (2 Pontos) Declare uma estrutura para representar uma Data. A estrutura deve conter os campos dia, mês e ano.
   (c) (2 Pontos) Declare uma estrutura para representar uma Atividade. A estrutura deverá conter os campos título, descrição, nota e Data de entrega.
   (d) (2 Pontos) Implemente uma função que receba (por referência) uma Atividade e (por valor) a Data de hoje. A função deverá alterar a nota da atividade para metade se a atividade estiver atrasada e imprimir a mensagem "Entregue com atraso". Caso contrário imprimir a mensagem "Entregue no prazo". A função deverá obedecer a assinatura:

```
void valida (Atividade * entregue, Data hoje);
```

   (e) (2 Pontos) Implemente uma função que receba um vetor de Atividades e a quantidade de Atividades realizadas. A função deverá calcular e retornar a nota média das Atividades (considere que todas as Atividades possuem o mesmo peso). A função deverá obedecer a assinatura:

```
float media (Atividade atividades[], int quantidade);
```

5. Listas encadeadas:
   (a) (5 Pontos) Descreva e compare as vantagens e desvantagens entre a utilização de vetores e listas encadeadas.

Considerando uma lista encadeada para armazenar números inteiros:

(b) (5 Pontos) Declare uma estrutura para representar um elemento da lista;

Considerando que as funções `inserir` e `remover` da lista encadeada já estão implementadas e seguem a assinatura:

```
Item *inserir(Item *lista, int info);
Item *remover(Item *lista, int v);
```

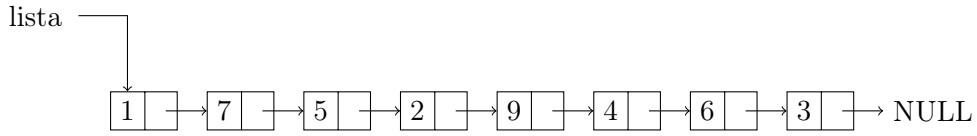
(c) (20 Pontos) Implemente uma função que receba como parâmetro uma lista encadeada. A função deverá obedecer a assinatura:

```
Item* separa (Item* lista);
```

Utilizando as funções `insere` e `remove` já implementadas, a função deverá:

i. Declarar uma nova lista vazia.
ii. Copiar todos os elementos pares para a nova lista
iii. Remover todos os elementos pares da lista original
iv. Retornar o endereço da nova lista

(d) (10 Pontos) Realize o teste de mesa da função implementada na letra c. Considere como valor de entrada a seguinte lista encadeada:



6. Pilhas e filas:

(a) (5 Pontos) Descreva as diferenças e semelhanças entre as estruturas de dados pilha e fila.

Considerando a implementação de pilha e fila utilizando vetores e que as funções de manipulação já estão implementadas e seguem a assinatura:

```
void enfileira(int fila[], int valor, int *inicio, int* qtd);
int desenfileira(int fila[], int *inicio, int *qtd);
void empilha (int pilha[], int valor, int* topo);
int desempilha (int pilha[], int* topo);
```

(b) (10 Pontos) Implemente uma função que receba como parâmetro uma fila. A função deverá obedecer a assinatura:

```
void calcula(int fila[], int *inicio, int *qtd);
```

A função deverá:

i. Declarar uma pilha vazia e inicializar o topo (-1).
ii. Desenfileirar valores da fila até que a fila esteja vazia.
iii. Para cada valor desenfileirado da fila:
   - Se o valor desenfileirado for 1 então desenfileira mais um valor da fila e empilha na pilha.
   - Se o valor desenfileirado for 2 então desempilha dois valores da pilha, soma os dois valores e empilha o resultado na pilha.
   - Se o valor desenfileirado for 3 então desempilha um valor e imprime na tela.

(c) (15 Pontos) Realize o teste de mesa da função implementada na letra b. Considere como valor de entrada a seguinte fila:

1	4	1	5	2	1	6	2	3
0	1	2	3	4	5	6	7	8