CSI 103 – Algoritmos e Estruturas de Dados I

1ª Lista de Exercícios

Prof. Mateus Ferreira Satler

DECSI – ICEA - UFOP

Obs. 1: para todos os exercícios de **codificação (criação de código na linguagem C)** a seguir, utilize a modularização em C, ou seja, crie arquivos separados **`.h`** e **`.c`**. Crie 1 projeto separado para cada exercício (não faça mais de um exercício dentro do mesmo código).

Obs. 2: os exercícios **3**, **4** e **5** **NÃO TRATAM DE GERAÇÃO DE CÓDIGO NA LINGUAGEM C**. Resolva os exercícios à mão em uma folha ou usando qualquer editor de texto de sua preferência. Gere **1 arquivo PDF PARA CADA EXERCÍCIO EM SEPARADO (NÃO COLOQUE TODOS OS EXERCÍCIOS EM UM MESMO PDF).** Os envios que estiverem em desacordo com esse padrão **SERÃO DESCONSIDERADOS.**

1. Considere um TAD **Circulo**, que oferece a seguinte interface, definida em um arquivo **`circulo.h`**:

```
typedef struct Circulo_Est {
   float x;
   float y;
   float r;
}Circulo;

/* Cria um circulo com centro (x,y) e raio r */
Circulo * circ_cria (float x, float y, float r);

/* Libera a memoria de um Circulo */
void circ_libera (Circulo * c);

/* Calcula a area de um circulo */
float circ_area (Circulo * c);

/* Verifica se dois circulos tem o mesmo centro */
int circ_concentricos (Circulo * c1, Circulo * c2);

/* Obtem o raio de um circulo */
float circ_raio (Circulo * c);
```

a) Construa uma **implementação** para este TAD, que deverá ser o conteúdo de um módulo **`circulo.c`**. Não esqueça de incluir nesse módulo a interface do TAD e outros arquivos de cabeçalho necessários.

b) Escreva também a função **`main`** que cria dois **`Circulos`** (usando a função **`circ_cria`**), testa se os dois círculos são concêntricos e, se forem, imprime o raio do maior deles. Imprima também a área dos dois círculos. No final, libere a memória dos dois círculos.

2. Escreva um algoritmo completo na linguagem C seguindo as instruções abaixo. Leia atentamente todos os itens antes de iniciar seu desenvolvimento:

a) Crie uma estrutura chamada livro. Cada livro terá os seguintes dados: título (150 caracteres), autor (150 caracteres), preços (vetor de números reais com 6 posições). O vetor "preços" representa o preço do livro nos últimos 6 meses.

b) Crie um procedimento que peça ao usuário para digitar todos os dados de um conjunto de livros. Os parâmetros do procedimento são: um vetor de livros e seu tamanho.

c) Crie um procedimento para imprimir o preço médio de cada livro do vetor de livros. O preço médio será calculado pela média aritmética de todos os preços de cada livro. Os parâmetros deste procedimento são: um vetor de livros e seu tamanho.

d) Faça uma função principal (main) para criar dinamicamente um vetor de 5 livros usando a estrutura do item (a) e, posteriormente, "chame" os procedimentos dos itens (b) e (c). Imprima também as informações do livro mais caro e do livro mais barato.

3. Analise a complexidade dos algoritmos a seguir:

a)
```
float func1(int n, float A[], float x)
{
  int k;
  float y = 0.0;
  for (k = n; k >= 0; k--)
  {
    y = A[k] + y * x;
  }
  return y;
}
```

b)
```
int func2(int n)
{
  int i, j, x, soma = 0;
  for (i = 0; i < n; i++)
  {
    for (j = 0; j < n; j++)
    {
      for (x = 0; x < n; x++)
      {
        soma += n;
      }
    }
  }
  return soma;
}
```

c)
```
void func3(int* A, int n)
{
  int i, j, aux;
  for( j = 2; j <= n; j++){
    aux = A[j];
    i = j - 1;
    while (i > 0 && A[i] > aux){
      A[i + 1] = A[i];
      i = i -1;
    }
    A[i + 1] = aux;
  }
}
```

4. Usando a definição formal de Θ, prove que $6n^3 \neq \Theta(n^2)$.

5. As funções ***log (n)*** e ***log(n^2)*** possuem a mesma ordem de complexidade? Justifique sua resposta.

6. Faça um procedimento recursivo que receba um inteiro N e que imprima na tela o valor de N invertido.
Ex: 123 – 321. Faça um pequeno programa para testar seu procedimento.

7. Crie uma função recursiva que receba um número inteiro positivo N e calcule o somatório dos números de 1 a N. Faça um pequeno programa para testar sua função.

8. Escreva uma função recursiva que procure um valor em um vetor e retorne o índice do elemento, caso ele exista no vetor, ou -1 caso, caso contrário. Faça um pequeno programa para testar sua função.

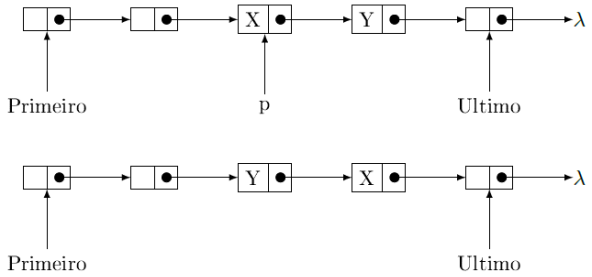
9. Crie uma função recursiva que calcula e retorna a soma de todos os elementos de um vetor de números reais de tamanho n. Faça um pequeno programa para testar sua função.

10. Escreva um procedimento na Linguagem C para trocar de posição os **elementos** **m** e **n** de uma lista implementada por **vetores** (**m** e **n** podem ser chaves ou mesmo ponteiros para os elementos, a escolha é sua). Faça um pequeno programa para testar seu procedimento.

11. Escreva um procedimento na Linguagem C para trocar de posição os elementos de **índice** **i** e **j** de uma lista implementada por **ponteiros – lista encadeada**. Faça um pequeno programa para testar seu procedimento.

12. Considere listas implementadas por **ponteiros – lista encadeada**. Escreva um procedimento **`inssub (l1, i1, l2, i2, len)`** para inserir os elementos da lista **`l2`**, começando no elemento de índice **`i2`** e continuando por **`len`** elementos na lista **`l1`**, começando na posição **`i1`**. Nenhum elemento da lista **`l1`** deverá ser removido ou substituído. A lista **`l2`** deve permanecer inalterada. Faça um pequeno programa para testar seu procedimento.

13. Considere a implementação de **listas encadeadas** utilizando **ponteiros**. Escreva um procedimento **`Troca(TipoLista * pLista, TipoCelula * p)`** que, dado um ponteiro para uma célula qualquer (**p**), troca de posição essa célula com a sua célula seguinte da lista, como mostrado na figura abaixo. (**Obs.:** Não vale trocar apenas o campo item! Você deverá fazer a manipulação dos ponteiros para trocar as duas células de posição). Não esqueça de tratar os casos especiais. Faça um pequeno programa para testar seu procedimento.



14. Existem partes de sistemas operacionais que cuidam da ordem em que os programas devem ser executados. Por exemplo, em um sistema de computação de tempo-compartilhado (“time-shared”) existe a necessidade de manter um conjunto de processos em uma fila, esperando para serem executados. Considere que cada processo é representado por um registro composto por um número identificador do processo (**`ID`**). Escreva um programa na Linguagem C que seja capaz de ler uma série de solicitações para:

- **a)** Incluir novos processos na fila de processos.
- **b)** Retirar da fila o processo com o maior tempo de espera (o primeiro a entrar na fila).
- **c)** Imprimir o conteúdo da lista de processo em determinado momento.
- **d)** Imprimir o comprimento (ou seja, o número de processos) da fila.

15. Implemente uma fila de inteiros em C, usando uma implementação por arranjos (um vetor **`queue[100]`**), onde **`queue[0]`** e **`queue[1]`** são usados para representar, respectivamente, a posição inicial e final da fila, e **`queue[2]`** a **`queue[99]`** são usados para armazenar os elementos do vetor. Demonstre como inicializar esse vetor de modo a representar a fila vazia e escreva funções **`Desenfileira`**, **`Enfileira`** e **`Vazia`** para tal implementação. Faça um pequeno programa para testar sua implementação.

16. Implemente um programa na linguagem C que contemple uma fila de contatos para um *call center*. As opções do programa devem ser:

- **a)** **Inserir Contato:** deve solicitar ao usuário os dados do cliente e incluir na fila. Os dados são: ID, nome completo e telefone.

b) **Próximo Contato:** deve pegar o contato do início da fila, removê-lo e mostrar os seus dados na tela para o usuário efetuar o contato com o cliente.
c) **Sair.**

17. Utilizando as operações de manipulação de pilhas vistas em sala, uma pilha auxiliar e uma variável do tipo **`TipoItem`**, escreva um procedimento que remove um item com chave **`c`** de uma posição qualquer de uma pilha. Note que você não tem acesso à estrutura interna da pilha (topo, item, etc.), apenas às operações de manipulação. Faça um pequeno programa para testar seu procedimento.

18. Escreva um algoritmo, usando Pilha, que inverte as letras de cada palavra de um texto terminado por ponto (.) preservando a ordem das palavras. Por exemplo, dado o texto:

`ESTE EXERCICIO E MUITO FACIL.`

A saída deve ser:

`ETSE OICICREXE E OTIUM LICAF`

19. Desenvolva um procedimento para inverter a posição dos elementos de uma pilha. Faça um pequeno programa para testar seu procedimento.

20. Desenvolva uma função para testar se duas pilhas P1 e P2 são iguais. Faça um pequeno programa para testar sua função.