**Estruturas de Dados**

**Exercícios**

**- Recursividade**

1) Faça um programa que apresente um menu inicial com as seguintes opções:

MENU PRINCIPAL

1 – Funções sem vetor

2 – Funções com vetor

Digite a opção desejada: \_

Se o usuário escolher a opção 1, solicite um número inteiro inicial e um número inteiro final. Em seguida apresente um segundo menu com as seguintes opções:

MENU SECUNDÁRIO

1 - Inteiros em ordem crescente

2 - Inteiros em ordem decrescente

3 - Inteiros ímpares (crescente)

4 - Somatório dos inteiros

Faça uma função recursiva para cada opção desse segundo menu.

Se o usuário escolher a opção 2 (do menu inicial), solicite vários números inteiros (armazene num vetor) e em seguida faça uma função recursiva que calcule o somatório dos números do vetor.

2) Escreva uma função recursiva chamada potencia(x, y), que retorne a base x elevado ao expoente y.

3) Implemente uma versão recursiva da seguinte função iterativa. Faça a repetição recursiva somente do for, não precisa fazer da expressão i \* i \* i.

void cubos (int n)

{

for (int i = 1; i <= n; i++)

Controle.WrilteLine(i \* i \* i);

}

4) Baseado no algoritmo de Euclides, implemente uma função recursiva para determinar o máximo divisor comum (mdc) entre dois números inteiros x e y.

Algoritmo de Euclides:

se (x = y) retorna x

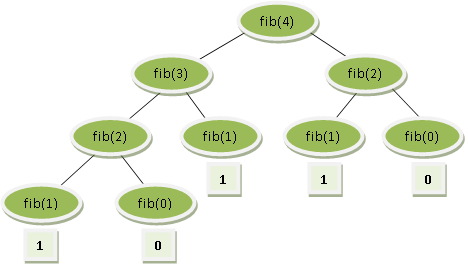
senão se (x < y) retorna mdc(y, x)

senão retorna mdc(x - y, y)

5) A Série de Fibonacci pode ser definida da seguinte maneira:

se (n == 0 ou n == 1) retorna n

se (n >= 2) retorna fib(n – 1) + fib(n – 2)



Seja fib(n) uma função que retorna o n-ésimo termo da série de Fibonacci, implemente uma versão recursiva e outra iterativa. Observe que a partir de um determinado número, a função recursiva começa a ficar mais lenta que a iterativa.

6) Defina uma função recursiva que converte um número inteiro para a base binária.

**- Lista linear sequencial: Pilha**

7) Ilustre a situação da pilha (memória), inicialmente vazia, após a execução de cada uma das operações a seguir:

1. Insere(pilha, ref topo, 20);
2. Insere(pilha, ref topo, 30);
3. Insere(pilha, ref topo, 40);
4. v = Remove(pilha, ref topo);
5. Insere(pilha, ref topo, 50);
6. Insere(pilha, ref topo, v);
7. Remove(pilha, ref topo);
8. Remove(pilha, ref topo);

8) Faça um programa que converta um número inteiro da base decimal para a base binária.

9) Escreva um programa para conhecer uma frase e exibi-la com as PALAVRAS invertidas. Por exemplo, a frase “Jose da Silva” deve sair “esoJ ad avliS”.

10) Elabore um programa que verifica se uma dada expressão é válida em relação aos abre e fecha parênteses, colchetes e chaves. Exemplos:

a) a = b + (c -d) \* (e – f) - válida

b) b = [c – d) - inválida

c) while (m < (n[8] + o)) { m = m + 1; } - válida

d) for (i = 1; i <= tl; i++ - inválida

11) Desenvolva um algoritmo para conversão de expressões da notação tradicional (in-fixa) para a notação polonesa (pós-fixa). As expressões são compostas somente por parênteses, operandos de apenas uma letra e os operadores de soma, subtração e multiplicação.

Como entrada de dados o usuário informa a expressão in-fixa a ser analisada, como saída de dados seu algoritmo deve apresentar a mesma expressão na notação pós-fixa.

Entrada

A+B

A+B–C

(A+B)\*(C–D)

(A+B\*C)

Saída

AB+

AB+C–

AB+CD-\*

ABC\*+

**- Lista linear sequencial: Fila**

12) Ilustre a situação da fila (memória), inicialmente vazia, após a execução de cada uma das seguintes operações:

1. Insere(fila, ref fim, 20);
2. Insere(fila, ref fim, 30);
3. Insere(fila, ref fim, 40);
4. Remove(fila, ref inicio);
5. Remove(fila, ref inicio);
6. Insere(fila, ref fim, 50);
7. Remove(fila, ref inicio);
8. Insere(fila, ref fim, 60);
9. Insere(fila, ref fim, 70);
10. v = Remove(fila, ref inicio)
11. Insere(fila, ref fim, v);

13) Escreva um programa que tenha uma fila cujo valores indicam prioridade, quanto maior o valor, maior a prioridade.

Seu programa deve inserir vários valores numa fila e solicitar um valor divisor. Em seguida, o programa deve dividir a fila inicial em duas novas filas, uma fila com valores menores que o divisor (menor prioridade) e outra fila com os valores maiores que o divisor (maior prioridade).

Por fim, informe os valores das duas novas filas.

14) Escreva um programa que insira vários números numa fila. Após a digitação dos números, seu programa deve encontrar o maior, o menor e a média aritmética dos números da fila. Por fim, informe os resultados encontrados.

15) Escreva um programa que simule o controle de uma pista de decolagem de aviões em um aeroporto. Os aviões são identificados pelos números digitados pelo usuário. Neste programa, o usuário deve ser capaz de realizar as seguintes operações:

a) Adicionar vários aviões à fila de espera para decolagem

b) Consultar a quantidade de aviões aguardando na fila

c) Autorizar a decolagem de um avião da fila

d) Listar os números de todos os aviões na fila

e) Consultar o número do primeiro avião da fila

Construa um menu principal para oferecer essas operações ao usuário.

**- Lista linear encadeada**

16) Ilustre a situação da lista encadeada (memória), inicialmente vazia, após a execução de cada uma das operações a seguir:

1. Insere(ref lista, 20);
2. Insere(ref lista, 30);
3. Insere(ref lista, 40);
4. v = Remove(ref lista);
5. v = Remove(ref lista);

17) Faça uma implementação que construa uma lista encadeada. Seu programa deve ter as opções de inserção e remoção dos elementos. Após remover um elemento da lista, exiba-o na tela.

18) Faça um programa que utilize lista encadeada e que tenha as opções a seguir. O nó deve conter os atributos: nome, idade, whats e prox.

1. Incluir conforme apresentado em aulas
2. Para alterar, consulte pelo nome. Se encontrar, exiba os valores atuais e permita a alteração. Caso não encontre, exiba mensagem de não encontrado.
3. Para excluir, procure pelo nome. Se encontrar, exiba os valores atuais e permita a exclusão. Caso não encontre, exiba mensagem de não encontrado.
4. Na opção exibir, exiba todos os registros.

19) Faça um programa para o usuário digitar vários números, inclua-os num vetor. Em seguida, faça uma função recursiva que copie os números do vetor para uma lista encadeada. Após copiar os números do vetor para a lista, percorra a lista e encontre o menor, o maior e a média dos números.

20) Implemente a lógica da inserção e remoção em uma lista duplamente encadeada (nó com ponteiro para o nó anterior e o próximo nó). Faça a procura pelo nó a ser excluído.

21) Implemente a lógica de uma lista encadeada com o conceito de fila, ou seja, insira os nós no fim da lista e remova-os do início da lista. Utilize duas variáveis (inicio e fim) para controlar os acessos a lista.

**- Lista não linear: Árvore Binária**

22) Seja uma árvore binária inicialmente vazia, ilustre seu estado final após terem sido inseridos os elementos a seguir, nesta ordem:

a) Insere(ref raiz, 4);

b) Insere(ref raiz, 1);

c) Insere(ref raiz, 0);

d) Insere(ref raiz, 5);

e) Insere(ref raiz, 3);

f) Insere(ref raiz, 7);

g) Insere(ref raiz, 2);

h) Insere(ref raiz, 6);

i) Insere(ref raiz, 9);

j) Insere(ref raiz, 8);

23) Considerando a árvore esquematizada a seguir, responda:

1. Quais são os nós folhas
2. O nível de cada nó não terminal
3. A altura da árvore
4. O grau da árvore
5. Os descendentes do nó c

24) Desenvolva um programa para inserir, pesquisar, remover e exibir os valores de uma árvore binária. Observe as opções a seguir:

1. Inserir um valor digitado pelo usuário
2. Pesquisar um valor digitado pelo usuário. Exiba uma mensagem informando se encontrou ou não
3. Remover um valor digitado pelo usuário. Exiba a mensagem se removido com sucesso ou não encontrado
4. Exibir todos os valores da árvore em ordem, pré ordem ou pós ordem

25) Baseado no atravessamento “em ordem” desenvolva:

a) um algoritmo para percorrer os nós e exibir apenas os valores armazenados nas folhas

b) uma rotina que determina se uma árvore é estritamente binária. OBS: Dizemos que uma árvore é estritamente binária se todo nó pertencente a ela é folha ou tem os dois filhos

c) um algoritmo para contar o número de nós não-terminais

**- Hash**

26) Ilustre a situação do vetor hash (memória), inicialmente vazio, após a execução de cada uma das operações a seguir. Considere um vetor para cada sequência de operações.

1. Insere(vetor, 22);
2. Insere(vetor, 40);
3. Insere(vetor, 34);
4. Insere(vetor, 44);
5. InsereLinear(vetor, 22);
6. InsereLinear (vetor, 40);
7. InsereLinear (vetor, 34);
8. InsereLinear (vetor, 44);
9. InsereEncadeado(vetor, 22);
10. InsereEncadeado (vetor, 40);
11. InsereEncadeado (vetor, 34);
12. InsereEncadeado (vetor, 44);

27) Implemente um programa que conte a quantidade de colisões ocorridas durante o processo de inserção. Utilize o tratamento de colisão linear. O vetor deve ser de um tipo abstrato de dado composto por nota, nome e email. A nota deve ser um número inteiro e corresponderá a chave.

O menu deve conter as seguintes opções: Inserir, Recuperar e Informar. Na opção recuperar, solicite a nota e caso encontre-a no vetor, exiba o nome e o email correspondente, caso contrário, exiba a mensagem de nota não encontrada. A opção informar, informa a quantidade de colisões ocorridas até o momento.

28) Implemente um programa com as seguintes opções: Sem tratamento de colisão, Tratamento de colisão Linear e Tratamento de colisão com Lista Encadeada.

Dentro de cada opção deve haver as funcionalidades: Inserir, Alterar e Relatar.

O vetor deve ser do tipo abstrato de dado composto por idade, nome e whats. Serão necessários 3 vetores, um para cada tipo de tratamento de colisão.

Para inserir um novo registro, solicite a idade, nome e whats. Utilize a idade como chave.

Para alterar solicite a idade (chave) para ser utilizada na busca. Caso encontrada, informe o nome e o whats da pessoa. Após a consulta, o usuário pode atualizar somente o nome e o whats.

Para relatar, percorra o vetor do inicio ao fim e exiba todos os registros.