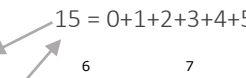


EXERCÍCIOS DE FIXAÇÃO Nº 01.8

Vetores e Matrizes com Funções (usando VLA)

1 - Dado um vetor de inteiros, preencha-o com somatórios.

O programa deve permitir que o usuário indique a quantidade de elementos do vetor (que deve ser maior que zero). Use uma função para preencher o vetor com o valor dos somatórios acumulados dos índices dos vetores, conforme ilustrado abaixo:



0	1	2	3	4	5	6	7	8	9	10	...
0	1	3	6	10	15	21	28	36	45	55	...

Após preencher o vetor com os somatórios, mostre os valores dos elementos na tela, também usando uma função. Use VLA, tanto na declaração do vetor quanto nas funções. Sugestões de protótipos para as funções:

```
void preenche_vetor_somatorio(int t,int vet[t]);  
void mostra_vetor(int t,int vet[t]);
```

2 - Dado um vetor de reais, faça a ordenação em ordem crescente ou decrescente.

O usuário poderá indicar a quantidade de elementos do vetor (que deve ser maior que zero). Os elementos do vetor devem ser preenchidos com valores randômicos. Os valores do vetor devem ser mostrados na tela usando uma função com VLA. Após isto, o usuário poderá indicar se quer a ordenação dos valores do vetor em ordem crescente ou decrescente. Use uma única função (também com VLA) para fazer a ordenação dos valores do vetor. Para isso, sugere-se o uso de um parâmetro adicional indicando a ordem ('c'=crescente e 'd'=decrescente). Ao final, mostre os valores do vetor após a ordenação. Sugestões de protótipos para as funções:

```
void preenche_vetor_rand(int t,int vet[t]);  
void mostra_vetor(int t,int vet[t]);  
void ordena_vetor(int t,int vet[t],char ordem);
```

Dica: considere a possibilidade de usar o operador ternário na função de ordenação

Ex. de operador ternário:

```
if( universitario == 'S' ? idade < 24 : idade < 18 )  
    printf("Dependente com direito a plano de saúde");
```

3 – Dado um vetor de inteiros, faça a busca binária de um valor (chave).

O programa deverá pedir que o usuário indique o número de elementos para o vetor. Use uma função para inserir valores aleatórios no vetor. Com outra função, mostre os valores na tela os valores aleatórios inseridos. Após isso, permita que o usuário digite um valor que deseja buscar no vetor. Escreva uma função que recebe o vetor v, sua capacidade n e a chave de busca. A função deve fazer a busca binária e indicar o índice no qual encontrou a chave. Lembre-se que a busca binária ocorre em um vetor ordenado e será necessário fazer a ordenação antes da busca. Sugestões de protótipos de funções:

```
void preenche_vetor_rand(int t,int vet[t]);  
void mostra_vetor(int t,int vet[t]);  
void ordena_vetor(int t,int vet[t],char ordem);  
int busca_binaria_chave(int t,int vet[t],int chave);
```

4 - Dada uma matriz original qualquer, gere a matriz transposta desta original.

A dimensão da matriz original (número de linhas e número de colunas) deve ser indicada pelo usuário. Ambos devem maiores que 1. Os valores da matriz original podem ser gerados randomicamente numa função. A matriz transposta deve ser uma outra matriz (não sobrescrever os valores da matriz original) e pode ser gerada em uma função. Os valores de ambas as matrizes devem ser mostrados na tela. Recomenda-se usar uma mesma função com VLA. Sugestões para os protótipos das funções:

```
void preenche_matriz_rand(int lin,int col,int mat[lin][col]);  
void mostra_matriz(int lin,int col,int mat[lin][col]);  
void gera_matriz_transposta(int lin,int col,int mat[lin][col],int transp[col][lin]);
```

5 – Dadas 2 matrizes A e B compatíveis, gerar a matriz C, produto de $A \times B$.

O programa deve ler as dimensões (número de linhas e número de colunas) tanto da matriz A quanto da matriz B. Se ambas as dimensões forem compatíveis para uma operação de multiplicação (número de colunas de A deve ser igual ao número de linhas de B), o programa deve prosseguir. Caso contrário, deve mostrar uma mensagem de erro para o usuário e encerrar a execução. Os valores das matrizes A e B devem ser gerados randomicamente para as duas matrizes(A e B). Deve-se usar uma mesma função para geração dos valores randômicos usando VLA. Após isso, mostrar os valores de ambas as matrizes (uma de cada vez) na tela. Também sugere-se usar uma única função com VLA. Em seguida, gerar a matriz C com o produto da matriz A pela matriz B. A matriz produto também deverá ser impressa na tela com a mesma função usada para imprimir as duas primeiras matrizes. Sugestões para os protótipos das funções:

```
void preenche_matriz_rand(int lin,int col,int mat[lin][col]);  
void mostra_matriz(int lin,int col,int mat[lin][col]);  
void produto_matrizes(int l1,int c1,int l2,int c2,int A[l1][c1],int B[l2][c2],int C[l1][c2]);
```