Exercício 3:

1. Pegar o número de elementos da lista (vetor) A;
2. Pegar os elementos, com base no dado anterior, da lista A;
3. Pegar o número de elementos da lista B;
4. Pegar os elementos da lista B;
5. Criar lista C com os elementos da lista A e a lista B;
6. Organizar, com a ajuda de um temporário, em ordem crescente, a lista C em ordem crescente;
7. Printar a lista C.

3) Intercalação é o processo utilizado para construir uma lista ordenada, de tamanho n+m, a partir de duas listas já ordenadas de tamanhos n e m. Por exemplo, a partir das listas:

A

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 3 | 6 | 7 |

B

|  |  |  |
| --- | --- | --- |
| 2 | 4 | 5 |

Construímos a lista C

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Fazer um programa que:

a) Leia NA, número de elementos do conjunto A (NA<=100);

b) Leia, em seguida, os elementos do conjunto A;

c) Leia, logo após o valor de NB, número de elementos do conjunto B (NB<=100);

d) Leia, finalmente, os elementos do conjunto B;

e) Crie e imprima um conjunto C, ordenado, de tamanho NA+NB, a partir dos conjuntos originais A e B.

Observações:

• Considerar os elementos de A e B como inteiros

• Considerar que os elementos de A e B já são lidos ordenados

------------------------------------------------------------------------------------------------------------------------

Programa em C:

#include<stdio.h>

int main (){

int NA;

int NB;

int A[100];

int B[100];

int C[200];

int i, j;

int n, g;

printf(“Programa criado para realizar intercalacao entre duas listas inseridas pelo usuario.\nInsira o numero de elementos que a lista A tera (entre 0 e 100): ”);

scanf(“%d”, &NA);

while ((NA<0)||(NA>100)){

printf(“Valor invalido para o numero de elementos de A.\n Insira novamente o numero de elementos da lista A: “);

scanf(“%d”, &NA);

}

printf(“Insira agora os elementos, de maneira ordenada, da lista A:\n“);

for(i=0; i<NA; i++){

printf(“Elemento %d de %d: “, i, NA);

scanf(“%d”, &A[i]);

}

printf(“Insira agora o numero de elementos que a lista B tera (entre 0 e 100): ”);

scanf(“%d”, &NB);

while ((NB<0)||(NB>100)){

printf(“Valor invalido para o numero de elementos de B.\n Insira novamente o numero de elementos da lista B: “);

scanf(“%d”, &NB);

}

printf(“Insira agora os elementos, de maneira ordenada, da lista B:\n“);

for(i=0; i<NB; i++){

printf(“Elemento %d de %d: “, i, NB);

scanf(“%d”, &B[i]);

}

g=0;

for(i=0; i<=NA; i++){

n = 0;

for(j=0; j<=NB; j++){

if((B[j]<=A[i])&&(j>n)){

C[g]=B[j];

n++;

g++;

}

}

if(A[i]<=C[(g-1)]){

C[g]=A[i];

g++;

}

}

for(i=0; i<g; i++){

printf(“%d\n”, C[i]);

}

system(“pause”);

return 0;

}