

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE

SÃO PAULO

EULLEN SILVA VIANA

JOÃO MARCELO

LABORATÓRIO DE PROGRAMAÇÃO II

EXPERIMENTO 5 - TAREFA EXTRA 5

* Descrição do Algoritmo

Algoritmo Intercalando Vetores

Objetivo: Ler dois vetores, em seguida ordená-los em ordem crescente e depois intercalá-los em um terceiro vetor mantendo a ordem.

Dados de Entrada: v, v2 (lista de reais);

Dados de Saída: v3 (lista de reais);

Pré condição: v[x] ≠ 0, v[x] ≠ 0 ;

intercalandoVetores();

**função lerVetor ( v[] ( real ), cont[] ( inteiro ) ) : sem retorno**

cont[0] ← -1 ; i ← 0;

repita

leia ( v[i]; cont ← cont + 1; i ← i + 1;

até que ( ( v[i-1] ≠ 0) e (cont[0] < 100) );

**função exibirVetor ( v[] ( real ) , TV ( inteiro ) ) : sem retorno**

para k de 0 até ( tv – 1 ) repita

imprima( (k+1) , v[k]);

**função ordenarVetor ( v[] ( real ) , cont ( inteiro ) ) : sem retorno**

t ← cont – 1;

para conta de 1 até ( cont – 1 ) repita

para j de 0 até ( t – 1) repita

se ( v[j] > v[j+1] ) então

temp ← v[j]; v[j] ← v[j+1]; v[j+1] ← temp;

t ← t – 1;

**função intercalarVetor ( v[], v2[], v3[] ( real ), cont, cont 2 (inteiro) ) : inteiro**

k ← 0; j ← 0; l ← 0; sentinela ← -1;

se ( v[k] = 0 ) então sentinela ← 0;

se (v2[j] = 0 ) então sentinela ← 1;

enquanto

anoBissexto ← FALSE;

enquanto ( ( sentinela ≠ 0 ) e ( sentinela ≠ 1 ) ) faça

se ( v[k] < v2[j] ) então

v3[l] ← v[k]; k ← k + 1;

senão se ( v[k] = v2[j] ) então

v3[l] ← v[k] ; v3[l+1] ← v2[j];

k ← k + 1; j ← j + 1; l ← l + 1;

senão se ( v[k] > v2[j] ) então

v3[l] ← v2[j]; j ← j + 1;

se ( k = cont ) então sentinela ← 0;

se ( k = cont2 ) então sentinela ← 1;

l ← l + 1;

se ( sentinela = 1) então

para i de l até ( cont + cont2 ) repita

v3[i] ← v[k]; k ← k +1;

se ( sentinela = 0 ) então

para i de l até ( cont + cont2 ) repita

v3[i] ← v2[j]; j ← j +1;

**função Principal ( vazio )**

função lerVetor (v, cont );

função lerVetor (v2, cont2 );

função ordenarVetor (v, cont[0]);

função ordenarVetor (v2, cont2[0]);

função intercalarVetor (v,v2,v3,cont[0],cont2[0]);

função exibirVetor( v3, ( cont[0] + cont2[0] ) );

* Desenho da Tela a ser Impressa:

Digite os componentes do vetor ou ‘0’ para encerrar.

Valor #1: xx.xx

Valor #2: xx.xx

.

.

Valor #α : xx.xx

Digite os componentes do vetor ou ‘0’ para encerrar.

Valor #1: xx.xx

Valor #2: xx.xx

.

.

Valor #α : xx.xx

1: Componente : xx.xx

2. Componente : xx.xx

.

.

α. Componente: xx.xx

* Código Fonte em C:

*#include<stdio.h>*

*#include<stdlib.h>*

*void lerVetor(float v[],int cont[])*

*{*

*float temp;*

*cont[0]=-1; // para não começão do 0*

*int i = 0;*

*printf("Digite as componentes do vetor: \n");*

*printf("ou 0 para encerrar\n");*

*do*

*{*

*printf("Valor #%d:", i+1);*

*scanf("%f", &v[i]);*

*cont[0]++;*

*i++;*

*}while( v[i-1]!=0 && cont[0] < 100 );*

*printf("\n");*

*}*

*void exibirVetor( float v[ ], int tv )*

*{*

*int k;*

*for( k = 0; k < tv; k++ )*

*{*

*printf("%d: ",k+1);*

*printf("componente:%.2f\n",v[ k ] );*

*}*

*printf("\n");*

*}*

*//Ordenando pelo método da Bolha*

*void ordenarVetor( float v[], int cont )*

*{*

*int conta, j, t = cont-1;*

*float temp;*

*for( conta = 1; conta < cont; conta++ )*

*{*

*for( j = 0; j < t; j++ )*

*{*

*if( v[j] > v[j+1] )*

*{*

*temp = v[j]; v[j] = v[j+1];*

*v[j+1] = temp;*

*}*

*}*

*t = t-1;*

*}*

*}*

*void intercalarVetor(float v[],float v2[],float v3[], int cont,int cont2)*

*{*

*int k=0,j=0,l=0,i,sentinela=-1;*

*// sentinela usada para para o loop while quando um dos vetores for varrido*

*if(v[k]==0) sentinela=0;*

*//checa se o vetor é vazio, se for ele nÃo entra no laço while*

*if(v2[j]==0) sentinela=1;*

*//checa se o vetor é vazio, se for ele nÃo entra no laço while*

*while(sentinela!=0 && sentinela!=1)*

*// o valor da sentinela deverá ser ou 0 ou 1 para depois varrer o vetor que ainda nao acabou de ser varrido*

*//o loop verificará se um valor é maior, menor ou igual a posição do outro vetor e acumular até que algum vetor seja varrido completamente*

*{*

*if(v[k]<v2[j])*

*{*

*v3[l]=v[k];*

*k++;//avançando o cursor do vetor 1*

*} else if(v[k]==v2[j])*

*{*

*v3[l]=v[k];*

*v3[l+1]=v2[j];*

*l++;*

*//avançando o cursor do vetor 3, pois se há igualdade o vetor ocupará 2 posições*

*k++; //avançando o cursor do vetor 1*

*j++; //avançando o cursor do vetor 2*

*}else if(v[k]>v2[j])*

*{*

*v3[l]=v2[j];*

*j++; //avançando o cursor do vetor 2*

*}*

*// se o cursor for igual ao contador(que tem o numero de posições do vetor) significa que o vetor foi varrido completamente*

*if(k==(cont)) sentinela=0;*

*if(j==(cont2)) sentinela=1;*

*l++; //avançando o cursor do vetor 3*

*}*

*if(sentinela==1)//Varrendo o vetor 1 pois o vetor 2 não foi varrido*

*{*

*for(i=l;i<=cont+cont2;i++) // a variavel i é acumulada em 'l' pois é aonde o vetor 3 vai receber a proxima posição*

*{*

*v3[i]=v[k];*

*k++;*

*}*

*}*

*if(sentinela==0) //Varrendo o vetor 2 pois o vetor 1 nao foi varrido*

*{*

*for(i=l;i<=cont+cont2;i++) // i deve ser menor que (cont+cont2) pois é o tamanho de alocação somando os vetores 1 e 2*

*{*

*v3[i]=v2[j];*

*j++;*

*}*

*}*

*}*

*void main()*

*{*

*float v[100],v2[100],v3[100+100];*

*int cont[1],cont2[1];*

*lerVetor(v,cont); //lendo primeiro vetor*

*lerVetor(v2,cont2); //lendo segundo vetor*

*ordenarVetor(v,cont[0]); //ordenando primeiro vetor*

*ordenarVetor(v2,cont2[0]); //ordenando segundo vetor*

*printf("\n"); //pula linha*

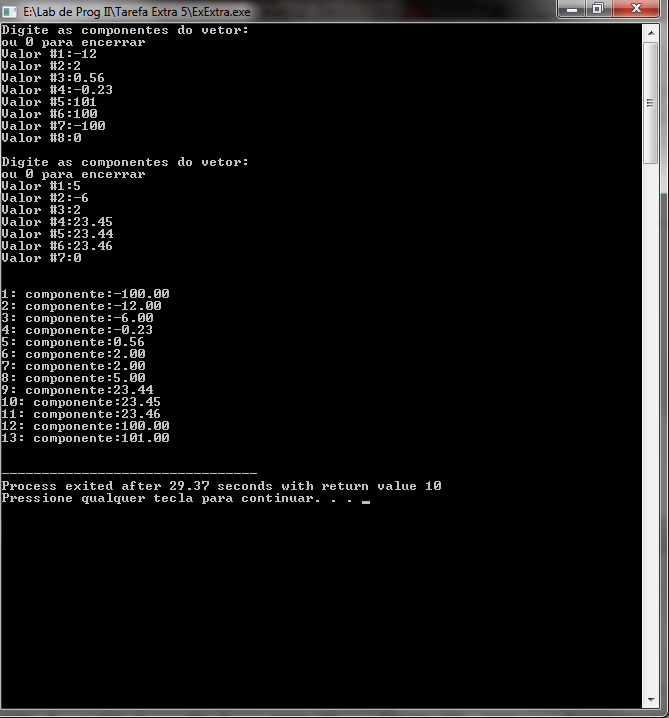
*intercalarVetor(v,v2,v3,cont[0],cont2[0]); //intercalando os vetores*

*exibirVetor(v3,(cont[0]+cont2[0])); //exibindo vetor intercalado*

*}*

* Testes

1)



2)

