CENTRO UNIVERSITARIO PARA O DESENVOLVIMENTO DO ALTO VALE DO ITAJAI

ACADEMICAS: ANA CRISTINA ZANETTE

CAROLINE KIESER

PROFESSOR: FERNANDO ANDRADE BASTOS

GRADUAÇÃO: BACHAREL EM SISTEMAS DE INFORMAÇÃO

***Trabalho Algoritmos e Estrutura de Dados I***

1. ***Faça um algoritmo para armazenar um vetor todos os divisores de um determinado numero que deve ser lido (informado). Escrever o vetor gerado.***

R:

PROGRAM EXERCICIO1;

USES CRT;

VAR CONT, N, I: INTEGER;

VET:ARRAY[1..50] OF INTEGER;

BEGIN

CLRSCR;

CONT:=0;

READLN (N);

FOR I:= 1 TO 50 DO BEGIN

IF ((N MOD I)=0) THEN BEGIN

VET[CONT]:=I;

CONT:=CONT+1;

END;

END;

FOR I:=1 TO CONT-1 DO

WRITELN (VET[I]);

READKEY;

END.

1. ***Faça um algoritmo que leia um vetor com 100 números inteiros, faça a troca do primeiro pelo ultimo, do segundo pelo penúltimo e assim sucessivamente e escreva o novo vetor.***

R:

PROGRAM EXERCICIO2;

USES CRT;

VAR I, AUX:INTEGER;

A:ARRAY[1..100] OF INTEGER;

BEGIN

CLRSCR;

FOR I:=1 TO 100 DO

READLN (A[I]);

FOR I:= 1 TO 50 DO BEGIN

AUX:=A[I];

A[I]:=A[51-I];

A[51-I]:=AUX;

END;

FOR I:=1 TO 100 DO

WRITELN (A[I]);

READKEY;

END.

1. ***Faça um algoritmo para ler 2 vetores: A e B, ambos com 100 elementos e informe quantos elementos de A>B, B>A e A=B comparando posição a posição.***

R:

PROGRAM EXERCICIO3;

USES CRT;

VAR I, IG, AM, BM: INTEGER;

A:ARRAY[1..100] OF INTEGER;

B:ARRAY[1..100] OF INTEGER;

BEGIN

CLRSCR;

IG:=0;

AM:=0;

BM:=0;

FOR I:=1 TO 100 DO BEGIN

WRITELN ('DIGITE VALORES PARA A');

READLN (A[I]);

WRITELN ('DIGITE VALORES PARA B');

READLN (B[I]);

IF A[I] = B[I] THEN BEGIN

IG:=IG+1;

END ELSE IF A[I] < B[I] THEN BEGIN

BM:=BM+1;

END ELSE

AM:=AM+1;

END;

WRITELN ('IGUAL: ',IG);

WRITELN ('A MAIOR: ', AM);

WRITELN ('B MAIOR: ', BM);

READKEY;

END.

1. ***Faça um algoritmo que leia um vetor com n elementos, em seguida verifique se um determinado elemento X qualquer se encontra no vetor. Em caso afirmativo, indicar a posição de X, caso contrario informar que o elemento não esta no vetor.***

R:

PROGRAM EXERCICIO4;

USES CRT;

VAR X, N, I:INTEGER;

A:ARRAY[1..10] OF INTEGER;

ACHOU, NAOACHOU:BOOLEAN;

BEGIN

CLRSCR;

READLN (N);

FOR I:=1 TO N DO

READLN (A[I]);

READLN (X);

I:=1;

ACHOU:=FALSE;

WHILE ((I<=10) AND (NAOACHOU)) DO BEGIN

IF A[I]=X THEN BEGIN

ACHOU:=TRUE;

I:=I+1;

END;

END;

IF ACHOU THEN BEGIN

WRITELN (X, 'ENCONTRADO');

END ELSE WRITELN ('NAO ENCONTRADO');

READKEY;

END.

***Pesquisa Seqüencial ou Busca Linear***

Forma mais simples de realizar pesquisas. Tem por metodologia percorrer o vetor, elemento por elemento, verificando se o elemento desejado está presente no vetor de modo que a função do tempo em relação ao número de elementos é linear, ou seja, cresce proporcionalmente.

Vantagem: Simplicidade

Desvantagem: Desempenho

Exemplo:

function procura(vetor :array [1..10] of integer; elementoProcurado: integer ; busca : boolean); {supondo que o vetor tem tamanho 10}

var

i : integer;

begin

busca:=false;

for i := 1 to 10 do

begin

if (vetor[i] = elementoProcurado) then

begin

procura := i; {retorna o índice do elemento procurado}

busca:=true; {para indicar que a busca encontrou o valor procurado no vetor}

end;

end;

end.

***Pesquisa Binária.***

A pesquisa ou busca binária é um [algoritmo de busca](http://pt.wikipedia.org/wiki/Algoritmo_de_busca) em [vetores](http://pt.wikipedia.org/wiki/Vetor) que requer [acesso aleatório](http://pt.wikipedia.org/wiki/Acesso_aleat%C3%B3rio) aos elementos do mesmo. Ela parte do pressuposto de que o vetor está ordenado e realiza sucessivas divisões do espaço de busca ([divisão e conquista](http://pt.wikipedia.org/wiki/Divis%C3%A3o_e_conquista)) comparando o elemento buscado (chave) com o elemento no meio do vetor. Se o elemento do meio do vetor for à chave, a busca termina com sucesso. Caso contrário, se o elemento do meio vier antes do elemento buscado, então a busca continua na metade posterior do vetor. E finalmente, se o elemento do meio vier depois da chave, à busca continua na metade anterior do vetor.

Vantagem : Simplicidade e desempenho

Desvantagem : Arquivo deve estar ordenado

Exemplo:

function BuscaBinaria (Vetor: array of string; Chave: string; Dim: integer): integer;

var inicio, fim: integer; {Auxiliares que representam o inicio e o fim do vetor analisado}

meio: integer; {Meio do vetor}

begin

fim := Dim; {O valor do último índice do vetor}

inicio := 1; {O valor do primeiro índice do vetor}

BuscaBinaria := -1; {Retorna o valor -1 se a chave nao for encontrada.}

repeat

meio := (inicio+fim) div 2;

if (Chave = vetor[meio]) then

begin

BuscaBinaria := meio;

inicio:=fim+1; {Interrompo o repeat quando a chave for encontrada sem ter que testar lá no until. Bonito não?!}

end;

if (Chave < vetor[meio]) then

fim:=(meio-1);

if (Chave > vetor[meio]) then

inicio:=(meio+1);

until (inicio > fim);