



Lista de Exercícios 04 – Estruturas de Dados Homogêneas - Vetores

- 5) Escreva um algoritmo em PORTUGOL que armazene em um vetor todos os números inteiros de 200 a 100 (em ordem decrescente). Após isso, o algoritmo deve imprimir todos os valores armazenados.

```
algoritmo L4P05;  
var  
  inteiro: C, VET[0..100];  
início  
  para C de 200 até 100 passo -1 faça  
    VET[200-C] <- C;  
  fim-para  
  para C de 0 até 100 faça  
    imprima(VET[C]);  
  fim-para  
fim
```

```
program l4p05;  
var  
  C: integer;  
  VET: array [0..100] of integer;  
begin  
  for C := 200 downto 100 do  
    VET[200-C] := C;  
  for C := 0 to 100 do  
    write(VET[C], ' ');  
  writeln('');  
end.
```

```
for C = 200 : -1 : 100  
  VET(200-C+1) = C;  
end  
for C = 0 : 100  
  fprintf(1, '%d ', VET(C+1));  
end  
fprintf(1, '\n');
```



- 10) Escreva um algoritmo em PORTUGOL que armazene em um vetor todos os números ímpares do intervalo fechado de 1 a 100. Após isso, o algoritmo deve imprimir todos os valores armazenados.

```
algoritmo L4P10;  
var  
  inteiro: C, VET[1..50];  
início  
  para C de 1 até 100 passo 2 faça  
    VET[C div 2 + 1] <- C;  
  fim-para  
  para C de 1 até 50 faça  
    imprima(VET[C]);  
  fim-para  
fim
```

```
program l4p10;  
var  
  C: integer;  
  VET: array [1..50] of integer;  
begin  
  for C := 1 to 100 do  
    if C mod 2 = 1 then  
      VET[C div 2 + 1] := C;  
  for C := 1 to 50 do  
    write(VET[C], ' ');  
    writeLn('');  
end.
```

```
for C = 1 : 2 : 100  
  VET( floor(C/2) + 1 ) = C;  
end  
for C = 1 : 50  
  fprintf(1, '%d ', VET(C));  
end  
fprintf(1, '\n');
```



- 15) Escreva um algoritmo em PORTUGOL que receba oito números do usuário e armazene em um vetor o logaritmo de cada um deles na base 10. Caso não seja possível calcular o valor para o número digitado, o número -1 deve ser atribuído ao elemento do vetor. Após isso, o algoritmo deve imprimir todos os valores armazenados.

```
algoritmo L4P15;
var
  inteiro: C;
  real: VALOR, LOGX[1..10];
inicio
  para C de 1 até 10 faça
    leia(VALOR);
    se ( VALOR > 0 ) então
      LOGX[C] <- log(VALOR) / log(10);
    senão
      LOGX[C] <- -1;
      imprima("Não é possível calcular o logaritmo! Número negativo ou zero!");
    fim-se
  fim-para
  para C de 1 até 10 faça
    imprima(LOGX[C]);
  fim-para
fim

program l4p15;
var
  C: integer;
  VALOR: real;
  LOGX: array [1..10] of real;
begin
  for C := 1 to 10 do
    begin
      write('Digite um valor: ');
      readln(VALOR);
      if ( VALOR > 0 ) then
        LOGX[C] := Ln(VALOR) / Ln(10)
      else begin
        LOGX[C] := -1;
        write('Nao eh possivel calcular o logartimo! Numero negativo ou zero!'); end;
      end;
    end;
  for C := 1 to 10 do
    write(LOGX[C], ' ');
  writeln('');
end.

for C = 1 : 10
  VALOR = input('Digite um valor: ');
  if ( VALOR > 0 )
    LOGX(C) = log10(VALOR); % ln(VALOR) / ln(10);
  else
    LOGX(C) = -1;
    disp('Nao eh possivel calcular o logartimo! Numero negativo ou zero!');
  end
end
for C = 1 : 10
  fprintf(1, '%f ', LOGX(C));
end
fprintf(1, '\n');
```

- 20) Construa um algoritmo em PORTUGOL para calcular a média de valores PARES e ÍMPARES, de 50 números que serão digitados pelo usuário. Ao final o algoritmo deve mostrar estas duas médias. O algoritmo deve mostrar também o maior número PAR digitado e o menor número ÍMPAR digitado. Esses dados devem ser armazenados em um vetor. Além disso, devem ser impressos os valores PARES maiores que a média PAR, bem como os valores ÍMPARES menor que a média ÍMPAR.

```

algoritmo L4P20;
var
  inteiro: VALOR, VETVAL[1..50], SOMAPAR, SOMAIMP, MAIORPAR, MENORIMP, C, CPAR, CIMP;
  real:   MEDIAPAR, MEDIAIMP;
início
  MAIORPAR <- 0;
  MENORIMP <- 0;
  SOMAPAR <- 0;
  SOMAIMP <- 0;
  CPAR <- 0;
  CIMP <- 0;
  para C de 1 até 50 faça
    leia(VALOR);
    VETVAL[C] <- VALOR;
    se ( VETVAL[C] mod 2 = 0 ) { é par } então
      se ( ( VETVAL[C] < MAIORPAR ) ou ( CPAR = 0 ) ) então
        MAIORPAR <- VETVAL[C];
      fim-se
      SOMAPAR <- SOMAPAR + VETVAL[C];
      CPAR <- CPAR + 1;
    senão
      se ( ( VETVAL[C] > MENORIMP ) ou ( CIMP = 0 ) ) então
        MENORIMP <- VETVAL[C];
      fim-se
      SOMAIMP <- SOMAIMP + VETVAL[C];
      CIMP <- CIMP + 1;
    fim-se
  fim-para
  se ( CPAR <> 0 ) então
    imprima("Maior par: ",MAIORPAR);
    MEDIAPAR <- SOMAPAR / CPAR;
    imprima("A media dos valores pares digitados eh: " ,MEDIAPAR);
    imprima("Valores PARES maiores que a media PAR");
    para C de 1 até 50 faça
      se ( VETVAL[C] mod 2 = 0 ) e ( VETVAL[C] > MEDIAPAR ) então
        imprima(VETVAL[C]);
      fim-se
    fim-para
  senão
    imprima("Não foi digitado valor par!")
  fim-se
  se ( CIMP <> 0 ) então
    imprima("Menor impar: ",MENORIMP);
    MEDIAIMP <- SOMAIMP / CIMP;
    imprima("A media dos valores impares digitados eh: ",MEDIAIMP);
    imprima("Valores IMPARES menores que a media IMPAR");
    para C de 1 até 50 faça
      se ( VETVAL[C] mod 2 = 1 ) e ( VETVAL[C] < MEDIAIMP ) então
        imprima(VETVAL[C]);
      fim-se
    fim-para
  senão
    imprima("Não foi digitado valor impar!")
  fim-se
fim
  
```



```
program l4p20b;
var
  VALOR, SOMAPAR, SOMAIMP, MAIORPAR, MENORIMP, C, CPAR, CIMP: integer;
  VETVAL: array [1..50] of integer;
  MEDIAPAR, MEDIAIMP: real;
begin
  MAIORPAR := 0;
  MENORIMP := 0;
  SOMAPAR := 0;
  SOMAIMP := 0;
  CPAR := 0;
  CIMP := 0;
  for C := 1 to 50 do
  begin
    write('Digite um valor: ');
    readln(VALOR);
    VETVAL[C] := VALOR;
    if ( VETVAL[C] mod 2 = 0 ) { é par } then begin
      if ( ( VETVAL[C] < MAIORPAR ) or ( CPAR = 0 ) ) then
        MAIORPAR := VETVAL[C];
      SOMAPAR := SOMAPAR + VETVAL[C];
      CPAR := CPAR + 1; end
    else begin
      if ( ( VETVAL[C] > MENORIMP ) or ( CIMP = 0 ) ) then
        MENORIMP := VETVAL[C];
      SOMAIMP := SOMAIMP + VETVAL[C];
      CIMP := CIMP + 1; end;
    end;
  end;
  if ( CPAR <> 0 ) then begin
    writeln('Maior par: ',MAIORIMP);
    MEDIAPAR := SOMAPAR / CPAR;
    writeln('A media dos valores pares digitados eh: ',MEDIAPAR);
    writeln('Valores PARES maiores que a media PAR');
    for C := 1 to 50 do
      if ( VETVAL[C] mod 2 = 0 ) and ( VETVAL[C] > MEDIAPAR ) then
        write(VETVAL[C], ' ');
        writeln(''); end
    else
      writeln('Não foi digitado valor impar!');
  end;
  if ( CIMP <> 0 ) then begin
    writeln('Menor impar: ',MENORIMP);
    MEDIAIMP := SOMAIMP / CIMP;
    writeln('A media dos valores impares digitados eh: ',MEDIAIMP);
    writeln('Valores IMPARES menores que a media IMPAR');
    for C := 1 to 50 do
      if ( VETVAL[C] mod 2 = 1 ) and ( VETVAL[C] < MEDIAIMP ) then
        write(VETVAL[C], ' ');
        writeln(''); end
    else
      writeln('Não foi digitado valor impar!');
  end;
end.
```



```
SOMAPAR = 0;
SOMAIMP = 0;
CPAR = 0;
CIMP = 0;
for C = 1 : 50
    VALOR = input('Digite um valor: ');
    VETVAL(C) = VALOR;
    if ( mod(VETVAL(C),2) == 0 ) % é par
        if ( ( VETVAL(C) < MAIORPAR ) | ( CPAR == 0 ) )
            MAIORPAR = VETVAL(C);
        end
        SOMAPAR = SOMAPAR + VETVAL(C);
        CPAR = CPAR + 1;
    else
        if ( ( VETVAL(C) > MENORIMP ) | ( CIMP == 0 ) )
            MENORIMP = VETVAL(C);
        end
        SOMAIMP = SOMAIMP + VETVAL(C);
        CIMP = CIMP + 1;
    end
end
if ( CPAR ~= 0 )
    fprintf(1,'Maior par: %d\n',MAIORPAR);
    MEDIAIMP = SOMAIMP / CIMP;
    fprintf(1,'A media dos valores pares digitados eh: %f\n',MEDIAIMP);
    disp('Valores PARES maiores que a media PAR');
    for C = 1 : 50
        if ( mod(VETVAL(C),2) == 0 ) & ( VETVAL(C) > MEDIAIMP )
            fprintf(1,'%d ',VETVAL(C));
        end
    end
    fprintf(1,'\n');
else
    disp('Não foi digitado valor par!');
end
if ( CIMP ~= 0 )
    fprintf(1,'Menor impar: %d\n',MENORIMP);
    MEDIAPAR = SOMAPAR / CPAR;
    fprintf(1,'A media dos valores impares digitados eh: %f\n',MEDIAIMP);
    disp('Valores IMPARES menores que a media IMPAR');
    for C = 1 : 50
        if ( mod(VETVAL(C),2) == 1 ) & ( VETVAL(C) < MEDIAIMP )
            fprintf(1,'%d ',VETVAL(C));
        end
    end
    fprintf(1,'\n');
else
    disp('Não foi digitado valor impar!');
end
```



25) Fazer um algoritmo em PORTUGOL que:

- Leia um conjunto de valores inteiros correspondentes a 80 notas dos alunos de uma turma, notas estas que variam de 0 a 10;
- Calcule a frequência absoluta e a frequência relativa de cada nota;
- Imprima uma tabela contendo os valores das notas (de 0 a 10) e suas respectivas frequências absoluta e relativa.

Observações:

- Frequência absoluta de uma nota é o número de vezes em que aparece no conjunto de dados;
- Frequência relativa é a frequência absoluta dividida pelo número total de dados;
- Utilizar como variável composta somente aquelas que forem necessárias.

```
algoritmo L4P25;
var
  inteiro: C, NOTA, FREQAB[0..10];
início
  para C de 0 até 10 faça
    FREQAB[C] <- 0;
  fim-para
  para C de 1 até 80 faça
    imprima("Digite a nota: ");
    leia(NOTA);
    se (NOTA = 0) então
      FREQAB[0] <- FREQAB[0] + 1;
    fim-se
    se (NOTA = 1) então
      FREQAB[1] <- FREQAB[1] + 1;
    fim-se
    se (NOTA = 2) então
      FREQAB[2] <- FREQAB[2] + 1;
    fim-se
    se (NOTA = 3) então
      FREQAB[3] <- FREQAB[3] + 1;
    fim-se
    se (NOTA = 4) então
      FREQAB[4] <- FREQAB[4] + 1;
    fim-se
    se (NOTA = 5) então
      FREQAB[5] <- FREQAB[5] + 1;
    fim-se
    se (NOTA = 6) então
      FREQAB[6] <- FREQAB[6] + 1;
    fim-se
    se (NOTA = 7) então
      FREQAB[7] <- FREQAB[7] + 1;
    fim-se
    se (NOTA = 8) então
      FREQAB[8] <- FREQAB[8] + 1;
    fim-se
    se (NOTA = 9) então
      FREQAB[9] <- FREQAB[9] + 1;
    fim-se
    se (NOTA = 10) então
      FREQAB[10] <- FREQAB[10] + 1;
    fim-se
  fim-para
  para C de 0 até 10 faça
    imprima("A frequência absoluta da nota ",C," eh ",FREQAB[C]);
    imprima("A frequência relativa da nota ",C," eh ",FREQAB[C] / 80);
  fim-para
fim
```



```
algoritmo L4P25B;  
var  
  inteiro: C, NOTA, FREQAB[0..10];  
início  
  para C de 0 até 10 faça  
    FREQAB[C] <- 0;  
  fim-para  
  para C de 1 até 80 faça  
    imprima("Digite a nota: ");  
    leia(NOTA);  
    se (NOTA >= 0) e (NOTA <= 10) então  
      FREQAB[NOTA] <- FREQAB[NOTA] + 1;  
    senão  
      imprima("Nota inválida!");  
    fim-se  
  fim-para  
  para C de 0 até 10 faça  
    imprima("A frequência absoluta da nota ",C," eh ",FREQAB[C]);  
    imprima("A frequência relativa da nota ",C," eh ",100 * FREQAB[C] / 80);  
  fim-para  
fim
```

```
program l4p25;  
var  
  C , NOTA: integer;  
  FREQAB : array [0..10] of integer;  
begin  
  for C := 0 to 10 do      {inicializa o vetor com zeros}  
    FREQAB[C] := 0;  
  for C := 1 to 80 do  
    begin  
      write('Digite a nota: ');  
      readln(NOTA);  
      if (NOTA = 0) then  
        FREQAB[0] := FREQAB[0] + 1;  
      if (NOTA = 1) then  
        FREQAB[1] := FREQAB[1] + 1;  
      if (NOTA = 2) then  
        FREQAB[2] := FREQAB[2] + 1;  
      if (NOTA = 3) then  
        FREQAB[3] := FREQAB[3] + 1;  
      if (NOTA = 4) then  
        FREQAB[4] := FREQAB[4] + 1;  
      if (NOTA = 5) then  
        FREQAB[5] := FREQAB[5] + 1;  
      if (NOTA = 6) then  
        FREQAB[6] := FREQAB[6] + 1;  
      if (NOTA = 7) then  
        FREQAB[7] := FREQAB[7] + 1;  
      if (NOTA = 8) then  
        FREQAB[8] := FREQAB[8] + 1;  
      if (NOTA = 9) then  
        FREQAB[9] := FREQAB[9] + 1;  
      if (NOTA = 10) then  
        FREQAB[10] := FREQAB[10] + 1;  
    end;  
  for C := 0 to 10 do  
    begin  
      writeln('A frequência absoluta da nota ',C,' eh ',FREQAB[C]);  
      writeln('A frequência relativa da nota ',C,' eh ',FREQAB[C] / 80);  
    end;  
end.
```




```
program l4p25b;
var
  C, NOTA: integer;
  FREQAB: array [0..10] of integer;
begin
  for C := 0 to 10 do
    FREQAB[C] := 0;
  for C := 1 to 80 do
    begin
      write('Digite a nota: ');
      readLn(NOTA);
      if (NOTA >= 0) and (NOTA <= 10) then
        FREQAB[NOTA] := FREQAB[NOTA] + 1
      else
        writeLn('Nota inválida!');
      end;
    for C := 0 to 10 do
      begin
        writeLn('A frequencia absoluta da nota ',C,' eh ',FREQAB[C]);
        writeLn('A frequencia relativa da nota ',C,' eh ',100 * FREQAB[C] / 80 : 6:2);
      end;
    end.

for C = 1 : 11      %Inicializa o vetor com zeros
  freqab(C) = 0;
end
for C = 1 : 80
  nota = input('Digite a nota: ');
  if (nota == 0)
    freqab(1) = freqab(1) + 1;
  elseif (nota == 1)
    freqab(2) = freqab(2) + 1;
  elseif (nota == 2)
    freqab(3) = freqab(3) + 1;
  elseif (nota == 3)
    freqab(4) = freqab(4) + 1;
  elseif (nota == 4)
    freqab(5) = freqab(5) + 1;
  elseif (nota == 5)
    freqab(6) = freqab(6) + 1;
  elseif (nota == 6)
    freqab(7) = freqab(7) + 1;
  elseif (nota == 7)
    freqab(8) = freqab(8) + 1;
  elseif (nota == 8)
    freqab(9) = freqab(9) + 1;
  elseif (nota == 9)
    freqab(10) = freqab(10) + 1;
  elseif (nota == 10)
    freqab(11) = freqab(11) + 1;
  end
end
for C = 1 : 11
  fprintf(1,'A frequencia absoluta da nota %d eh %d\n',C-1,freqab(C));
  fprintf(1,'A frequencia relativa da nota %d eh %6.2f\n',C-1,100 * freqab(C) / 80);
end
```



```
for C = 1 : 11
    FREQAB[C] = 0;
end
for C = 1 : 80
    NOTA = input('write('Digite a nota: ');
    if ( (NOTA >= 0) & (NOTA <= 10) )
        FREQAB[NOTA+1] := FREQAB[NOTA+1] + 1;
    else
        disp('Nota inválida!');
    end
end
for C := 0 to 10 do
    fprintf(1,'A frequencia absoluta da nota %d eh %d',C,FREQAB[C+1]);
    fprintf(1,'A frequencia relativa da nota %d eh %6.2f',C,100 * FREQAB[C+1] / 80);
end
```

30) Seja

$$P = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0$$

Escrever um algoritmo em PORTUGOL que:

- Leia o valor de n, sendo $n \leq 20$;
- Leia os coeficientes a_i , $i = 0, 1, 2, \dots, n$;
- Calcule o valor de P para 10 valores lidos para x;
- Imprima o valor de x e o valor de P correspondente.

```
algoritmo L4P30;  
var  
  inteiro: C, I, J, N;  
  real: P, XEXP, X[1..10], COEFA[0..20];  
início  
  imprima("Digite o valor de n: ");  
  leia(N);  
  se ( ( N < 1 ) ou ( N > 20 ) ) então  
    imprima("erro");  
  senão  
    para C de 0 até n faça  
      imprima("Digite o coeficiente a",C, ": ");  
      leia(COEFA[C]);  
    fim-para  
    para C de 1 até 10 faça  
      imprima("Digite o ",C, ". valor de x: ");  
      leia(X[C]);  
    fim-para  
    para C de 1 até 10 faça  
      P <- COEFA[0];  
      para I de 1 até N faça  
        Xexp <- 1;  
        para J de 1 até I faça  
          XEXP := XEXP * X[C];  
        fim-para  
        P <- P + COEFA[I] * XEXP;  
      fim-para;  
    imprima("A soma P de x",C," eh: ",P);  
  fim-para  
fim-se  
fim
```



```
program l4p30;
var C, I, J, N: integer;
    P, Xexp: real;
    X: array [1..10] of real;
    coefA: array [0..20] of real;
begin
    write('Digite o valor de n: ');
    readln(N);
    if ( ( N < 1 ) or ( N > 20 ) ) then
        writeln('erro')
    else
        begin
            for C := 0 to N do
                begin
                    write('Digite o coeficiente a',C,': ');
                    readln(coefA[C]);
                end;
            for C := 1 to 10 do
                begin
                    write('Digite o ',C, '. valor de x: ');
                    readln(X[C]);
                end;
            for C := 1 to 10 do
                begin
                    P := coefA[0];
                    for I := 1 to N do
                        begin
                            Xexp := 1;
                            for J := 1 to I do
                                Xexp := Xexp * X[C];
                            P := P + coefA[I] * Xexp;
                        end;
                    writeln('A soma P de x',C, ' eh: ',P:10:4);
                end;
            end;
        end;
end.
```

```
N = input('Digite o valor de n: ');
for C = 0 : N
    fprintf(1,'Digite o coeficiente a%d',C);
    coefA(C+1) = input(': ');
end
for C = 1 : 10
    fprintf(1,'Digite o %d',C);
    X(C) = input('. valor de x: ');
end
for C = 1 : 10
    P = coefA(1);
    for I = 1 : N
        Xexp = 1;
        for J = 1 : I
            Xexp = Xexp * X(C);
        end
        P = P + coefA(I+1) * Xexp;
    end
    fprintf(1,'A soma P de x %d eh: %f\n',C,P);
end
```



35) Faça um algoritmo em PORTUGOL que leia uma variável composta de N valores numéricos ($N \leq 20$) e ordene essa variável em ordem crescente. O programa também deve ler um número k e imprimir, antes e depois da ordenação, o k-ésimo termo da variável composta.

```
algoritmo L4P35;  
var  
  inteiro: C, I, K, N;  
  real: AUX, VET[1..20];  
início  
  imprima("Digite o valor de N: ");  
  leia(N);  
  imprima("Digite os valores numericos da variavel composta: ");  
  para C de 1 até N faça  
    leia(VET[C]);  
  fim-para  
  imprima("Digite o valor de K: ");  
  leia(K);  
  imprima("O k-esimo termo antes da ordenacao eh: ",vetor[K]);  
  para C de 1 até N - 1 faça  
    para I de 1 até N - C faça  
      se (VET [I] > VET[I + 1]) então  
        AUX <- VET[I];  
        VET[I] <- VET[I + 1];  
        VET[I + 1] <- AUX;  
      fim-se  
    fim-para  
  fim-para  
  imprima("O k-esimo termo antes da ordenacao eh: ",VET[K]);  
fim
```

```
program l4p35;  
var C, I, K, N : integer;  
    AUX : real;  
    VETOR: array [1..20] of real;  
begin  
  write('Digite o valor de N: ');  
  readln(N);  
  writeln('Digite os valores numericos da variavel composta: ');  
  for C := 1 to N do  
    readln(VETOR[C]);  
  write('Digite o valor de K: ');  
  readln(K);  
  writeln('O k-esimo termo antes da ordenacao eh: ',VETOR[K]:6:3);  
  for C := 1 to N - 1 do  
    for I := 1 to N - C do  
      if (VETOR[I] > VETOR[I + 1]) then  
        begin  
          AUX := VETOR[I];  
          VETOR[I] := VETOR[I + 1];  
          VETOR[I + 1] := AUX;  
        end;  
      writeln('O k-esimo termo depois da ordenacao eh: ',VETOR[K]:6:3);  
    end.  
end.
```



```
N = input('Digite o valor de N: ');
fprintf(1,'Digite os valores numericos da variavel composta: \n');
for C = 1 : N
    VETOR(C) = input('');
end
K = input('Digite o valor de K: ');
fprintf(1,'O k-esimo termo antes da ordenacao eh: %f\n',VETOR(K));
for C = 1 : N - 1
    for I = 1 : N - C
        if (VETOR(I) > VETOR(I + 1))
            AUX = VETOR(I);
            VETOR(I) = VETOR(I + 1);
            VETOR(I + 1) = AUX;
        end
    end
end
fprintf(1,'O k-esimo termo depois da ordenacao eh: %f',VETOR(K));
```