Instituto Federal de Educação, Ciência e Tecnologia – Campus Cuiabá “Octayde Jorge da Silva”

Aluno: Vitor Bruno de Oliveira Barth

Exercícios da Aula de 02/02/2016

4 - Escreva um algoritmo que leia um vetor de 10 posições e mostre-o. Use uma subrotina que inverta o vetor, trocando o 1º elemento com o último, o 2º com o penúltimo e assim sucessivamente. Mostre o novo vetor depois da troca.

```
algoritmo
        defina TAM = 10
        declare vetor[TAM] literal
        declare i numérico

        para i de 0 a TAM-1 faça
                leia vetor[i]
        fim-para

        para i de 0 (TAM/2)-1 faça
                troca(&vetor[], i)
        fim-para
fim-algoritmo

subrotina troca(*vetor[] literal, i numérico)
        declare aux
        aux = vetor[i]
        vetor[i] = vetor[(TAM-1)-i]
        vetor[(TAM-1)-i] = aux
fim-subrotina
```

```c
#include <stdio.h>

void troca(int* vetor, int i);

void main() {
    int vetor[10], i;

    for(i = 0; i < 10; i++) {
        printf("Insira o valor de vetor[%i]", i);
        scanf("%i", &vetor[i]);
    }

    for(i =0; i < 4; i++)
        troca(vetor, i);

    for(i = 0; i < 10; i++)
        printf("%i ", vetor[i]);

        printf("\n");
}

void troca(int* vetor, int i) {
     int aux = vetor[i];
     vetor[i] = vetor[9-i];
     vetor[9-i] = aux;
}
```

5 - Faça um algoritmo que use uma subrotina para receber como parâmetro dois vetores, contendo valores da coordenada x e valores da coordenada y de pontos no plano cartesiano. A subrotina deve calcular os coeficientas a e b de uma reta u=ax+b que é a regressao linear dos pontos

$$a = \frac{n * \sum xy - \sum x * \sum y}{n\sum x^2 - (\sum x)^2} \qquad b = \frac{\sum y * \sum x^2 - \sum x * \sum xy}{n\sum x^2 - (\sum x)^2}$$

```
algoritmo
    declare vetorX[], vetorY[], i, tamanho numericos
    leia tamanho

    para i de 0 a tamanho-1
      leia vetorX[i]
      leia vetorY[i]
    fim-para

    calcular(vetorX, vetorY, tamanho)
fim-algoritmo

subrotina calcular(vetorX[], vetorY[], tamanho numéricos)
   declare somaXY, somaX, somaY, somaX2 numericos

   para i de 0 a tamanho-1
        somaXY = somaXY+(vetorX[i] * vetorY[i])
        somaX = somaX + vetorX[i]
        somaX2 = somaX2 + pow(vetorX[i], 2)
        somaY = somaY + vetorY[i]
   fim-para

     a = ((tamanho*somaXY)-(somaX*somaY))/((b*somaX2)-pow(somaX, 2))
     b = ((somaY*somaX2)-(somaX*somaXY))/((n*somaX2)-pow(somaX, 2))

     imprima a, b
fim-subrotina
```

```c
#include <stdio.h>
#include <math.h>
#define TAMANHO 10

void calcular(float* vetorX, float* vetorY);

void main() {
    float vetorX[TAMANHO], vetorY[TAMANHO];
    int i;

     for(i = 0; i < TAMANHO; i++) {
        printf("Insira o valor de X, Y para p%i(x, y)",
        scanf("%f, %f", &vetorX[i], &vetorY[i]);
    }

    calcular(vetorX, vetorY);
}

void calcular(float* vetorX, float* vetorY) {
    float somaXY, somaX, somaY, somaX2, a, b, tamanho;
        int i = 0;

    for (i = 0; i < TAMANHO; i++) {
        somaXY = somaXY+(vetorX[i] * vetorY[i]);
        somaX = somaX + vetorX[i];
        somaX2 = somaX2 + pow(vetorX[i], 2);
        somaY = somaY + vetorY[i];
    }


    a = ((TAMANHO*somaXY)-(somaX*somaY));
    a = a/((TAMANHO*somaX2)-pow(somaX,2));
    b = ((somaY*somaX2)-(somaX*somaXY));
    b = b/((TAMANHO*somaX2)-pow(somaX, 2));

    printf("Na regressão linear do plano cartesiano apresentado
            dada pela fórmula y=ax+b\nOs coeficientes a = %f e b = %f", a , b);
}
```

5 – Faça um algoritmo que

a) Leia um conjunto de linhas contendo, cada uma, o numero de um empregado, a hora de inicio (horas, minutos, segundos) e a hora de termino (horas, minutos, segundos) de uma determinada tarefa. A última linha (FLAG) conterá o numero do empregado negativo
b) Calcule para cada empregado, a duração da tarefa que ele executou, num mesmo dia, utilizando as duas subrotinas anteriormente definidas.
c) Escreva, para cada empregado, o seu número e a duração da tarefa em horas, minutos e segundos

```
algoritmo
    declare numero[], entrada[], saida[], duracao[], i, j numérico
    i = 0
    faça
        leia numero[i]
        se (numero > 0) então
            leia entrada[i] {formato HHMMSS, 24horas}
            leia saida[i] {formato HHMMSS, 24horas}
            i++
            convertaHorasEmSegundos(&entrada[i])
            convertaHorasEmSegundos(&saida[i])
        senão
            numero[i]=0
    enquanto (numero[i] != 0)
    para j de 0 a i-1 faça
        duracao[i] = saida[i]-entrada[i]
        imprima numero[i],  convertaSegundosEmHoras(duracao[i])
    fim-para
fim-algoritmo

subrotina convertaHorasEmSegundos(*horario numérico)
    declare horas, minutos, segundos numérico
    segundos = horario%1000
    *horario = horario-segundos
    minutos = (horario%10000)/100
    *horario = horario-minutos
    horas = (horario/10000)
    *horario = (horas*3600) + (minutos*60) + segundos
fim-algoritmo

subrotina convertaSegundosEmHoras(horario numérico)
    declare horas, minutos, segundos numérico
    horas = (inteiro) horario/3600
    horario = horario - (horas*3600)
    minutos = (inteiro)horario/60
    segundos = horario – (minutos*60)
    retorne horas, minutos, segundos
fim-subrotina
```

```c
#include <stdio.h>
#define TAM 100

void convertaHorasEmSegundos(int *horario);
void convertaSegundosEmHoras(int horario);

void main(){
    int numero[TAM], entrada[TAM], saida[TAM], duracao[TAM], i, j;
    i = 0;
    do {
        printf("Para sair digite o numero do funcionario negativo");
        printf("\nInsira o numero do funcionario: ");
        scanf("%i", &numero[i]);
        if (numero[i] > 0) {
            printf("Insira a hora de inicio (HHMMSS): ");
            scanf("%i", &entrada[i]);
            printf("Insira a hora de termino (HHMMSS): ");
            scanf("%i", &saida[i]);
            convertaHorasEmSegundos(&entrada[i]);
            convertaHorasEmSegundos(&saida[i]);
       }
        i++;
    } while (numero[i-1] > 0);
        i -= 1;
    printf("Funcionario\tInicio\t\tFinal\t\tDuracao\n");
    for (j = 0; j < i; j++) {
        duracao[j] = saida[j]-entrada[j];
        printf("%i\t\t", numero[j]);
        convertaSegundosEmHoras(entrada[j]);
        printf("\t\t");
        convertaSegundosEmHoras(saida[j]);
        printf("\t\t");
        convertaSegundosEmHoras(duracao[j]);
        printf("\n");
    }
}

void convertaHorasEmSegundos(int *horario) {
    int horas, minutos, segundos;
    segundos = (*horario)%100;
        *horario -= segundos;
        minutos = (*horario%10000)/100;
        *horario -= minutos*100;
        horas = (*horario/10000);
        *horario = (horas*3600) + (minutos*60) + segundos;
}

void convertaSegundosEmHoras(int horario) {
    int horas, minutos, segundos;
    horas = horario/3600;
        horario -= horas*3600;
        minutos = horario/60;
        segundos = horario-minutos*60;
    printf("%ih %im %is", horas, minutos, segundos);
}
```