

MAC 2166 – Introdução à Computação para Engenharia

ESCOLA POLITÉCNICA – PRIMEIRO SEMESTRE DE 2006

Prova Substitutiva – 3 de julho de 2006

1. (valor 2 pontos)

Simule a execução do programa abaixo, **destacando** a sua saída. A saída do programa consiste em tudo que resulta dos comandos printf's e o número digitado pelo usuário. O dado de entrada (variável `nusp`) é o seu número USP.

```
# include <stdio.h>                                     /* NA LINHA ABAIXO USE O SEU No. USP */
                                                         scanf ("%d", &nusp);
                                                         printf ("nusp = %d\n", nusp);

void f0 (int v[4], int n)
{
    int i;
    for (i = 0; i < 4; i++)
        v[i] = n - i;
}

void f1 (int a[2][2], int v[4])
{
    int i, j;
    for (i = 0; i < 2; i++)
        for (j = 0; j < 2; j++)
            a[i][j] = v[2*i + j];
}

int f2 (int k, int x)
{
    k = k + 1;
    x = x + 1;
    return k;
}

int f3 (int k, int *x)
{
    k = k + 1;
    *x = *x + 1;
    return k;
}

int main()
{
    int nusp, i, j, n;
    int a[2][2];
    int w[4];
    float z;

    printf ("Entre com seu no. USP: ");

    /* NA LINHA ABAIXO USE O SEU No. USP */
    scanf ("%d", &nusp);
    printf ("nusp = %d\n", nusp);

    n = nusp % 10;
    n = n % 4 + 6;
    printf ("1: nusp=%d n=%d\n", nusp, n);

    f0(w,n);

    printf("2: %d %d %d %d\n",
           w[0], w[1], w[2], w[3]);

    f1(a,w);

    printf("3: %d %d %d %d\n",
           a[0][0], a[0][1], a[1][0], a[1][1]);

    i = 0;
    j = 0;
    i = f2(j,w[j]);

    printf("4: i=%d j=%d n=%d\n",i, j, n);
    printf("5: %d %d %d %d\n",
           w[0], w[1], w[2], w[3]);

    i = 0;
    j = 0;
    i = f3(a[1][1],&w[0]);

    printf("6: i=%d j=%d n=%d\n",
           i, j, n);
    printf("7: %d %d %d %d\n",
           w[0], w[1], w[2], w[3]);
    printf("8: %d %d %d %d\n",
           a[0][0], a[0][1], a[1][0], a[1][1]);

    i=(2.0 * n + 1) / 2;
    z=(2 * n + 1) / 2;
    printf("9: %d %f \n", i, z);
    return 0;
}
```

SOLUÇÃO

Entre com seu no. USP: nusp = 0

1: nusp=0 n=6
2: 6 5 4 3
3: 6 5 4 3
4: i=1 j=0 n=6
5: 6 5 4 3
6: i=4 j=0 n=6
7: 7 5 4 3
8: 6 5 4 3
9: 6 6.000000

Entre com seu no. USP: nusp = 1

1: nusp=1 n=7
2: 7 6 5 4
3: 7 6 5 4
4: i=1 j=0 n=7
5: 7 6 5 4
6: i=5 j=0 n=7
7: 8 6 5 4
8: 7 6 5 4
9: 7 7.000000

Entre com seu no. USP: nusp = 2

1: nusp=2 n=8
2: 8 7 6 5
3: 8 7 6 5
4: i=1 j=0 n=8
5: 8 7 6 5
6: i=6 j=0 n=8
7: 9 7 6 5
8: 8 7 6 5
9: 8 8.000000

Entre com seu no. USP: nusp = 3

1: nusp=3 n=9
2: 9 8 7 6
3: 9 8 7 6
4: i=1 j=0 n=9
5: 9 8 7 6
6: i=7 j=0 n=9
7: 10 8 7 6
8: 9 8 7 6
9: 9 9.000000

Entre com seu no. USP: nusp = 4

1: nusp=4 n=6
2: 6 5 4 3
3: 6 5 4 3
4: i=1 j=0 n=6
5: 6 5 4 3
6: i=4 j=0 n=6
7: 7 5 4 3
8: 6 5 4 3
9: 6 6.000000

Entre com seu no. USP: nusp = 5

1: nusp=5 n=7
2: 7 6 5 4
3: 7 6 5 4
4: i=1 j=0 n=7
5: 7 6 5 4
6: i=5 j=0 n=7
7: 8 6 5 4
8: 7 6 5 4
9: 7 7.000000

Entre com seu no. USP: nusp = 6

1: nusp=6 n=8
2: 8 7 6 5
3: 8 7 6 5
4: i=1 j=0 n=8
5: 8 7 6 5
6: i=6 j=0 n=8
7: 9 7 6 5
8: 8 7 6 5
9: 8 8.000000

Entre com seu no. USP: nusp = 7

1: nusp=7 n=9
2: 9 8 7 6
3: 9 8 7 6
4: i=1 j=0 n=9
5: 9 8 7 6
6: i=7 j=0 n=9
7: 10 8 7 6
8: 9 8 7 6
9: 9 9.000000

Entre com seu no. USP: nusp = 8

1: nusp=8 n=6
2: 6 5 4 3
3: 6 5 4 3
4: i=1 j=0 n=6
5: 6 5 4 3
6: i=4 j=0 n=6
7: 7 5 4 3
8: 6 5 4 3
9: 6 6.000000

Entre com seu no. USP: nusp = 9

1: nusp=9 n=7
2: 7 6 5 4
3: 7 6 5 4
4: i=1 j=0 n=7
5: 7 6 5 4
6: i=5 j=0 n=7
7: 8 6 5 4
8: 7 6 5 4
9: 7 7.000000

2. (valor 8 pontos)

Uma famosa competição mundial de futebol na Alemanha reúne 32 times. Imagine que cada time é representado por um código que é um número inteiro de 0 a 31 (por exemplo, o código da Alemanha poderia ser 0, o da Argentina 1, o do Brasil 2, etc).

Para ganhar essa competição, são considerados os seguintes dados de cada time:

- PG - número de pontos ganhos (3 por vitória, 1 por empate, 0 por derrota)
- GM - número de gols marcados
- GS - número de gols sofridos (gols difíceis de marcar)
- V - número de vitórias
- S - saldo de gols (GM – GS para os não futebolistas)
- GA - gol average (igual a GM/GS para GS \neq 0 e igual a zero para GS = 0)

Considere nesta questão que foi definida uma constante MAX através de

```
#define MAX 100
```

item (a) (vale 2 pontos)

Escreva uma função em C com o seguinte protótipo:

```
void Classifica (int m, int Resultados[MAX][4], int Times[32][4]);
```

que recebe uma tabela `Resultados` contendo o resultado de m jogos, e calcula os dados PG, GM, GS e V para cada time, atualizando a tabela `Times`. Cada resultado é representado na forma $(t1, t2, n1, n2)$ cuja interpretação é a seguinte: no jogo $t1 \times t2$ o resultado foi $n1 \times n2$.

Exemplo: (0, 1, 2, 1) indica que o time 0 ganhou do time 1 por 2×1 .

item (b) (vale 1 ponto)

Escreva uma função de protótipo

```
int AchaMaiorPG (int Times[32][4]);
```

que devolve o código de um time com o maior número de pontos ganhos.

item (c) (vale 2 pontos)

Escreva um programa em C que leia um inteiro m , com $m \leq \text{MAX}$, os resultados de m jogos e imprima a classificação de cada um dos 32 times em ordem decrescente de número de pontos ganhos. Para cada time devem ser impressos os valores de PG, GM, GS, V, S e GA correspondentes. Use obrigatoriamente a função do item (a) mesmo que você não a tenha feito. Se houverem times com o mesmo número de pontos ganhos, qualquer ordem serve. Por exemplo, para os resultados abaixo com $n = 3$ (início da competição onde apenas os times 0, 1 e 2 jogaram entre si):

```
0 1 2 1
0 2 1 2
1 2 3 4
```

A classificação do campeonato seria a seguinte:

Time	PG	GM	GS	V	S	GA
2	6	6	4	2	2	1.50
0	3	3	3	1	0	1.00
1	0	4	6	0	-2	0.67

Dica: repetidamente, depois de encontrar e imprimir os dados de um time com maior número de pontos ganhos, altere o número de pontos ganhos para um valor negativo.

Sinta-se a vontade para escrever qualquer função que desejar.

SOLUCAO:

```
/*
*****
  Solução da Q2a, Q2b e Q2c da PSub de MAC2166 - 1o Semestre 2006
*****
*/

#include <stdio.h>
#include <stdlib.h>

#define MAX 100

#define T1 0
#define T2 1
#define G1 2
#define G2 3

#define PG 0
#define GM 1
#define GS 2
#define V 3

/* Protótipos */
void classifica (int m, int Resultados[MAX][4], int Times [32][4]);
int achaMaiorPG (int Times[32][4]);

/*****/
/* Solucao do item a */
/*****/
void classifica (int m, int Resultados[MAX][4], int Times [32][4])
{
    int i;
    int time1; /* número do time 1 */
    int time2; /* número do time 2 */
    int gols_de_T1; /* gols do time 1 */
    int gols_de_T2; /* gols do time 2 */

    /* inicializar tabela Times */
    for (i = 0; i < 32; i++) {
        Times[i][PG] = 0;
        Times[i][GM] = 0;
        Times[i][GS] = 0;
        Times[i][V] = 0;
    }

    /* calcular dados da classificacao a partir dos Resultados */
    for (i = 0; i < m; i++ ) {
        time1 = Resultados[i][T1];
        time2 = Resultados[i][T2];
        gols_de_T1 = Resultados[i][G1];
        gols_de_T2 = Resultados[i][G2];
        Times[time1][GM] = Times[time1][GM] + gols_de_T1;
        Times[time1][GS] = Times[time1][GS] + gols_de_T2;
        Times[time2][GM] = Times[time2][GM] + gols_de_T2;
        Times[time2][GS] = Times[time2][GS] + gols_de_T1;
        if (gols_de_T1 == gols_de_T2) {
            /* empate */

```

```

        Times[time1][PG] = Times[time1][PG] + 1;
        Times[time2][PG] = Times[time2][PG] + 1;
    }
    else if (gols_de_T1 > gols_de_T2) {
        Times[time1][V] = Times[time1][V] + 1;
        Times[time1][PG] = Times[time1][PG] + 3;
    }
    else {
        Times[time2][V] = Times[time2][V] + 1;
        Times[time2][PG] = Times[time2][PG] + 3;
    }
}
}

/*****
/* Solucao do item b */
*****/
int achaMaiorPG (int Times[32][4]) {
    int i;
    int cod = 0;
    for (i=0; i<32; i++)
        if (Times[i][PG] > Times[cod][PG])
            cod = i;
    return cod;
}

/*****
/* Solucao do item c */
*****/
int main()
{
    int i, m, maior;
    int Times[MAX][4];
    int Jogos[MAX][4];
    float ga = 0;

    printf("Digite o numero de jogos: ");
    scanf("%d", &m);

    printf("Digite os resultados de todos os jogos:\n");
    for (i=0; i<m; i++) {
        scanf("%d %d %d %d", &Jogos[i][T1], &Jogos[i][T2], &Jogos[i][G1], &Jogos[i][G2]);
    }

    classifica(m, Jogos, Times);

    printf("A classificacao do campeonato foi a seguinte: \n");
    printf("Time \t PG \t GM \t GS \t V \t S \t GA \n");
    /* imprime a classificacao maior PG primeiro */
    for (i=0; i<32; i++) {
        maior = achaMaiorPG (Times);
        ga = 0;
        if (Times[maior][GS] > 0)
            ga = (float) Times[maior][GM] / Times[maior][GS];
        printf("%d \t %d \t %d \t %d \t %d \t %d \t %d \t %4.2f\n",
            maior, Times[maior][PG], Times[maior][GM], Times[maior][GS], Times[maior][V],
            Times[maior][GM] - Times[maior][GS], ga);
        Times[maior][PG] = -1;
    }
    system("pause");
    return 0;
}

```

item (d) (vale 3 pontos)

Um grupo de torcedores organizou um bolão sobre os resultados dos m jogos. Cada resultado certo vale 5 pontos (inclusive o placar) ou 3 pontos (apenas o vencedor ou empate). Seu programa deve ler o seguinte:

- um inteiro m (número de jogos), com $m \leq \text{MAX}$, e um inteiro n (número de apostadores), com $n \leq \text{MAX}$;
- os resultados de m jogos;
- para cada um dos n participantes, seu programa deve ler m pares de números inteiros, onde o par representa o resultado de um jogo, dados na mesma ordem dos resultados.

Seu programa deve calcular e imprimir os pontos de cada apostador, e também imprimir os números dos ganhadores (caso houver mais de um, então imprimir todos).

Por exemplo:

Para a entrada:

```
3 4
0 1 2 1
0 2 1 2
1 2 3 4
1 1 1 2 2 0
0 0 0 1 0 1
1 1 0 5 0 5
0 3 0 0 6 0
```

Saida:

Os pontos de cada apostador sao:

```
apostador 0 : 5
apostador 1 : 6
apostador 2 : 6
apostador 3 : 0
```

0(s) ganhador(es) eh(sao):

```
apostador 1 : 6
apostador 2 : 6
```

Sinta-se à vontade para escrever qualquer função que desejar.

SOLUCAO:

```
/*
   Solução do item d da questão 2 da PSub de MAC2166 - 1o Semestre 2006
*/

#include <stdio.h>
#include <stdlib.h>

#define MAX 100

#define T1 0
#define T2 1
#define G1 2
#define G2 3

int main()
{
    int i, m, n, maior, jogo, g1, g2, apostador;
```

```

int Jogos[MAX][4];
int Pontos[MAX];

printf("Digite o numero de jogos e o numero de apostadores: ");
scanf("%d %d", &m, &n);

printf("Digite os resultados de todos os jogos:\n");
for (i=0; i<m; i++) {
    scanf("%d %d %d %d", &Jogos[i][T1], &Jogos[i][T2], &Jogos[i][G1], &Jogos[i][G2]);
}

/* inicializa vetor */
for (apostador=0; apostador < n; apostador++) Pontos[apostador] = 0;

/* le e calcula os pontos de cada apostador */
for (apostador=0; apostador<n; apostador++) {
    printf("Digite os palpites do apostador %d: \n", apostador);
    for (jogo = 0; jogo < m; jogo++) {
        scanf("%d %d", &g1, &g2);
        if (g1 == Jogos[jogo][G1] && g2 == Jogos[jogo][G2])
/* apostador acertou em cheio !! */
Pontos[apostador] += 5;
        else if (g1 == g2 && Jogos[jogo][G1] == Jogos[jogo][G2])
/* acertou empate */
Pontos[apostador] += 3;
        else if (g1 > g2 && Jogos[jogo][G1] > Jogos[jogo][G2])
/* acertou vitoria do time 1 */
Pontos[apostador] += 3;
        else if (g1 < g2 && Jogos[jogo][G1] < Jogos[jogo][G2])
/* acertou vitoria do time 2 */
Pontos[apostador] += 3;
    }
}

printf("Os pontos de cada apostador sao:\n");
for (apostador=0; apostador<n; apostador++)
    printf("apostador %d : %d \n", apostador, Pontos[apostador]);

/* ganhadores: primeiro acho maximo */
maior = Pontos[0];
for (apostador=1; apostador<n; apostador++)
    if (Pontos[apostador] > maior) maior = Pontos[apostador];
/* imprime os sortudos */
printf("O(s) ganhador(es) eh(sao):\n");
for (apostador=0; apostador<n; apostador++)
    if (Pontos[apostador] == maior)
        printf("apostador %d : %d \n", apostador, Pontos[apostador]);

system("pause");
return 0;
}

```