

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
INTRODUÇÃO À CIÊNCIA DA COMPUTAÇÃO - ICC
Turma B - 2a. Prova – 22/6/2010

RESOLUÇÃO

Obs: Os programas mostrados nesta resolução estão completos. Na prova, procurou-se considerar versões mais reduzidas e simplificadas.

1. Seja a seguinte definição:

```
struct caixa
{
    char construtor[40];
    float altura;
    float largura;
    float comp;
    float volume;
};
```

- a) escreva uma função `f1` que recebe o endereço de uma estrutura `caixa` e que faz com que o campo `volume` seja igual ao produto das outras três dimensões (1 ponto);
- b) escreva uma função `f2` que recebe uma estrutura `caixa` por valor e que mostre o valor de cada campo (0.5 ponto);
- c) escreva uma função `main()` que chame `f1` e `f2` (0.5 ponto).

Solução:

```
a)
void f1(struct caixa *c1)
{
    c1 -> volume = (c1 -> altura) * (c1 -> largura) * (c1 -> comp);
    printf("%g\n", c1 -> volume);
}

b)
void f2 (struct caixa c2)
{
    printf("\n%s\n%g\n%g\n%g\n%g\n\n", c2.construtor, c2.altura, c2.largura,
c2.comp, c2.volume);
}

c)
#include <stdio.h>

main()
{
    struct caixa cx = {
        "Abelhudo",
        (float) 12.3,
        (float) 2.3,
        (float) 0.22,
        (float) 2.32
    };

    f1 (&cx);

    f2 (cx);
}
```

2. A função de Akermann é definida da seguinte maneira:

$$\begin{aligned}
 A(m,n) &= n + 1, & \text{se } m = 0 \\
 &= A(m-1,1), & \text{se } m > 0 \text{ e } n = 0 \\
 &= A(m-1, A(m,n-1)) & \text{se } m > 0 \text{ e } n > 0
 \end{aligned}$$

Escreva uma função recursiva que recebe inteiros não negativos m e n e devolve $A(m, n)$.

Solução:

```
#include <stdio.h>

int akermann(unsigned m, unsigned n);

main()
{
    unsigned m, n;
    printf("Funcao de Akermann\n\n");
    printf("Entre com os valores de m e n: ");
    scanf("%d%d", &m, &n);
    printf("O valor de Akermann %c %d\n", 130, akermann(m,n));
}

int akermann(unsigned m, unsigned n)
{
    if (m == 0)
        return n + 1;
    else
        if (n == 0)
            return akermann(m-1,1);
        else
            return akermann(m-1, akermann(m,n-1));
}
```

3. Escreva o que e como serão impressos os resultados do programa abaixo:

```
#include <stdio.h>

void p (int *x, int y, int *z);

main()
{
    int a,b,c;
    a = 5; b = 8; c = 3;
    p(&a,b,&c);
    printf("a = %d, b = %d, c = %d\n",a,b,c);
    p(&c,b,&a);
    printf("a = %d, b = %d, c = %d\n",a,b,c);
    p(&b,a+b+c,&c);
    printf("a = %d, b = %d, c = %d\n",a,b,c);
}

void p(int * x, int y, int * z)
{
    *z = *x+y-*z; *x = *x+*z; y = y-*x;
}
```

Solução:

```
a = 15, b = 8, c = 10
a = 3, b = 8, c = 13
a = 3, b = 27, c = 19
```

4. Fazer um programa para ler uma string STR qualquer da entrada e fazer a sua inversão. Por exemplo, se STR for "ANTONIO CARLOS" deverá ser gerado o seu inverso que é "SOLRAC OINOTNA". A string original deve ser mantida inalterada e o programa deverá ser composto de duas funções:

- Uma função **main** que lê STR da entrada, exibe o seu conteúdo, chama a função de conversão e exibe o resultado devolvido pela função (string invertida);
- Uma função **inverte** que recebe uma string na lista de parâmetros, determina o seu inverso e retorna-o à função chamadora através de um comando **return**.

Solução: `#include <stdio.h>`

```
char * inverte (char * str)
{
    int i, j;
    char s1[20];
    char * s = s1;
    for (i = strlen(str)-1, j = 0; i >= 0; i--, j++)
        s[j] = str[i];
    s[strlen(str)] = '\0';
    return s;
}

main()
{
    char st[20], si[20];
    printf("Entre com a string: ");
    gets(st);
    strcpy(si, inverte(st));
    printf("String original = %s\nString invertida = %s\n", st, si);
}
```

5. Escreva um programa que leia uma matriz 10 x 10, troque a coluna C com a coluna D (deverão ser lidos valores para C e D) e imprima a matriz resultante. Antes de efetuar a troca das colunas, testar se C é diferente de D e (C e D) maiores ou iguais a zero e menores do que 9.

Solução: `#include <stdio.h>`

```
main()
{
    int mat1[10][10];
    int i, j, C, D, temp;
    printf("\nEntre com o conteudo da matriz\n");
    for (i = 0; i < 10; i++)
        for (j = 0; j < 10; j++)
        {
            printf("\nEntre com o conteudo de [%d][%d]: ", i, j);
            scanf("%d", &mat1[i][j]);
        }
    printf("\nEntre com a coluna C: ");
    scanf("%d", &C);
    printf("\nEntre com a coluna D: ");
    scanf("%d", &D);
    if ((C != D) && (C >= 0) && (C < 10) && (D >= 0) && (D < 10))
        for (i = 0; i < 10; i++)
        {
            temp = mat1[i][C];
            mat1[i][C] = mat1[i][D];
            mat1[i][D] = temp;
        }
    printf("\nMatriz resultante\n\n");
    for (i = 0; i < 10; i++)
    {
        for (j = 0; j < 10; j++)
            printf("%d\t", mat1[i][j]);
        printf("\n");
    }
}
```