

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
INTRODUÇÃO À CIÊNCIA DA COMPUTAÇÃO - ICC
Turma D - 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. Escreva uma função, denominada `concatena`, que recebe como parâmetros dois vetores numéricos (*que sejam capazes de armazenar quaisquer números*), e seus respectivos tamanhos. A função deve gerar um vetor que contém a concatenação dos dois vetores passados como parâmetros e imprimir seu conteúdo. Assuma 20 como tamanho máximo deste vetor resultante. Escreva a função principal `main()` com a chamada da função `concatena`.

Solução:

```
#include <stdio.h>

void concatena(float n1[], int t1, float n2[], int t2);

main()
{
    float vet1[3] = {(float) 1.11, (float) 2.22, (float) 3.33};
    float vet2[3] = {(float) 4.44, (float) 5.55, (float) 6.66};

    concatena(vet1, 3, vet2, 3);
}

void concatena(float n1[], int t1, float n2[], int t2)
{
    float vet3[20];
    int i;

    for(i = 0; i < t1; i++)
        vet3[i] = n1[i];

    for(i = 0; i < t2; i++)
        vet3[i+t1] = n2[i];

    for(i = 0; i < (t1+t2); i++)
        printf("%g\n", vet3[i]);
}
```

2. 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(&b,a+b+c,&c);
    printf("a = %d, b = %d, c = %d\n",a,b,c);
    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);
}

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

Solução:

```
a = 5, b = 29, c = 21
a = 18, b = 29, c = 13
a = 24, b = 29, c = 37
```

3. Escreva um programa que leia uma string (adote tamanho máximo 80 e permita digitação de espaços) e imprima o total de palavras da string lida e o número de palavras que se iniciam com os caracteres 'A' ou 'a'. Considere que entre cada palavra existe apenas um espaço.

Exemplo: Entrada = "A estrutura atômica do Alumínio é simples".
Saídas: Total de palavras = 7, Total de palavras iniciadas com 'A' = 3.

Solução:

```
#include <stdio.h>

main()
{
    char str[80];
    short pal_a = 0;
    short tot_pal = 0;
    int i = 0;

    printf("Entre com uma string: ");
    gets(str);
    do
    {
        if (str[i] == 'A' || str[i] == 'a')
            pal_a++;
        while ((str[i] != ' ') && (i < strlen(str)))
            i++;
        tot_pal++;
        i++;
    } while (i < strlen(str));

    printf("String Lida = %s\n", str);
    printf("Total de palavras = %d\n", tot_pal);
    printf("Total de palavras iniciadas com 'A' = %d\n", pal_a);
    getch();
}
```

4. Considere a tabela com as distâncias entre algumas cidades do exercício 10 da Lista de Exercícios do capítulo 6:

<i>Origem/Destino</i>	São Carlos	Araraquara	Rio Claro	São Paulo
São Carlos	0	40	70	200
Araraquara	40	0	110	240
Rio Claro	70	110	0	130
São Paulo	200	240	130	0

Suponha que, além das distâncias, a tabela contenha também a informação da existência de pedágio ('S') ou não ('N') entre as cidades. Qual é o melhor tipo de dados para armazenar esta informação juntamente com a distância? Por quê? Como você faria para verificar se há pedágio entre duas cidades?

Solução:

```
struct
{
    int dist;
    char pedagio;
} M[4][4];
```

```
if (M[indice(c1[0],c1[4])][indice(c2[0],c2[4])].pedagio == 'S')
    printf("ha pedagio.\n");
else
    printf("Nao ha pedagio.\n");
```

5. O valor de x^n pode ser definido recursivamente como: $x^0 = 1$, $x^n = x * x^{n-1}$. Escreva uma função recursiva que calcule e retorne o valor de x^n .

Solução:

```
#include <stdio.h>
int xis(int, int);

main()
{
    int x, n;

    printf("Entre com o valor de x: ");
    scanf("%d", &x);
    printf("Entre com o valor de n: ");
    scanf("%d", &n);

    printf("O valor de %d elevado a %d %c %d\n\n", x, n, 130, xis(x,n));
}

int xis(int a, int b)
{
    if (b > 0)
        return a * xis(a,b-1);
    else
        return 1;
}
```