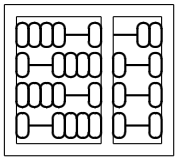
MC 102 - Algoritmos e programação de computadores
Instituto de Computação — UNICAMP

Quinta lista de exercícios



1. Execute os programas a seguir no papel, mantendo uma tabela com todos os valores que as variáveis dos programas assumem.

```
#include <stdio.h>

void flat();
void side();

int main()
{
     int i;

      flat();
      for (i=1; i<=6; i++) side();
      flat();

      return 0;
}

void flat()
{
      int i;

      for (i=1; i<= 10; i++) printf("*");
      printf("\n");

      return ;
}

void side()
{
    int i;

    printf("*");
    for (i=1; i<= 8; i++) printf(" ");
    printf("*\n");

    return ;
}
```

```
#include <stdio.h>

void P(int x, int y, int *pz);

int main()
{
    int a, b, c;

    a = 5; b = 8; c = 3;
    P(a,b,&c);
    P(7,a+b,&c);
    P(a*b, a/b, &c);

    return 0;
}

void P(int x, int y, int *pz)
{
    *pz = x + y + *pz;
    printf("%d\t%d\t%d\n", x, y, *pz);

    return ;
}
```

```c
#include <stdio.h>

void P(int i, int j, int *pk);
void R(int *pi);
void Q(int h, int *pj);

int main()
{
    int i, j, k;

    i = 0; j = 1; k = 2;
    Q(0,&k); Q(1, &i); Q(2, &j);

    return 0;
}

void P(int i, int j, int *pk)
{
    *pk = *pk + 1;
    printf("%d\t%d\t%d\n", i, j, *pk);

    return ;
}

void R(int *pi)
{
    *pi = *pi+1;

    return ;
}

void Q(int h, int *pj)
{
    int i;

    i = *pj;
    if (h==0) P(i, h, pj);
    else if (h==1) P(h, *pj, &i);
    else R(&i);
    printf ("%d\t%d\t%d\n",i, *pj, h);

    return ;
}
```

```c
#include <stdio.h>

void S(int *ps, int *pq, int *pr);
void T(int *ps, int *pt);

int main()
{
    int x=90, y=25, z=50;

    S(&x, &y, &z);
    printf("%d\t%d\t%d\n",x,y,z);
    x=6; y=2; z=1;
    S(&x, &y, &z);
    printf("%d\t%d\t%d\n",x,y,z);

    return 0;
}

void T(int *ps, int *pt)
{
    int *paux;

    paux = ps; ps = pt; pt = paux;

    return ;
}


void S(int *ps, int *pq, int *pr)
{
    if (*ps > *pq) T(ps, pq);
    if (*pq > *pr) {
        T(pq,pr);
        if (*ps > *pq) T(ps,pq);
    }

  return ;
}
```

2. Faça uma função que receba como parâmetro um vetor de inteiros e retorne, *simultaneamente*, o índice do maior elemento e o índice do menor elemento do vetor.

3. Faça uma função que inverta os elementos de um vetor. Sua função **não** deve usar outros vetores além do vetor recebido como parâmetro.

4. Faça uma função que converta uma seqüência de dígitos para a sua representação em código Morse. A representação dos dígitos em código Morse está explicitada na tabela abaixo.

1	. - - - -	6	- . . . .
2	. . - - -	7	- - . . .
3	. . . - -	8	- - - . .
4	. . . . -	9	- - - - .
5	. . . . .	0	- - - - -

5. Faça uma função que calcule a aproximação para a integral:

$$\int_0^x e^{-u^2}\,du = x - \frac{x^3}{3*1!} + \frac{x^5}{5*2!} - \frac{x^7}{7*3!} + \cdots$$

Sua função deverá ter o seguinte protótipo:

double aprox_int(**float** x, **float** cte),

onde x é o mesmo x da fórmula acima e *cte* é um valor utilizado como critério de parada da aproximação, isto é, a sua função deverá interromper o cálculo quando o i-ésimo termo, em valor absoluto, ficar menor que *cte*.

6. Faça uma função que receba duas strings e substitua todas as ocorrências da segunda string na primeira string por "*". Por exemplo, se a primeira string for "abcdabcdbbacdbba" e a segunda string "cd" a primeira string deve ser modificada para "ab*ab*bba*bba".

7. Suponha que tenham sido feitas as seguintes declarações:

#define MAX 20

typedef int[MAX][MAX] Tpmatriz

Escreva uma função que receba uma matriz como parâmetro e retorne, simultaneamente: (i) o maior elemento da matriz; (ii) o menor elemento da matriz (iii) o elemento que mais se aproxima de seu valor médio.

8. Em uma eleição presidencial existem quatro candidatos. Os votos são informados através dos seguintes códigos:

1, 2, 3, 4	⇒	votos para os respectivos candidatos;
5	⇒	voto nulo;
6	⇒	voto em branco.

Suponha que os votos tenham sido armazenados em um vetor. Faça um programa que simule uma eleição (carregando o vetor de votos) e posteriormente calcule as seguintes informações:

(a) Total de votos para cada candidato, de votos brancos e nulos.

(b) Porcentagem de votos para cada candidato sobre o total de votos válidos (sem considerar brancos e nulos.

(c) Porcentagem de votos nulos sobre o total de votos;

(d) Porcentagem de votos brancos sobre o total de votos.

Seu programa deve ter as seguintes funções:

float totaliza(**int** votos[], **int** codigo) e
float porcentagem(**int** total1, **int** total2).

A função *totaliza* recebe o vetor de votos e um código e retorna o número de votos com aquele código. A função *porcentagem* recebe o número de ocorrências de um conjunto de dados e o total sobre o qual a porcentagem deve ser calculada. Os dados solicitados devem ser calculados usando-se as funções *totaliza* e *porcentagem*.

9. Faça uma função que receba como parâmetros um vetor de inteiros e o tamanho lógico do vetor e modifique este vetor de maneira que todos os números divisíveis pr 2 ocorram antes dos números não divisíveis por 2.

10. Escreva uma função que receba um número inteiro $0 \leq n \leq 1.000.000$ e imprima o seu valor pr extenso.

11. Escreva uma função

int num_occorrencias(**int** vet[], **int** l, **int** u, **int** x)

que retorna o número de ocorrências do elemento x no vetor $v[l, \ldots, u]$.

12. Faça uma função

int teste(**int** dia, **int** mes, **int** ano)

e retorne: (i) 1 se a data é válida; e (ii) 0 se a data não existe.

13. Escreva uma função que receba um vetor como parâmetro e retorne: (i) 1, se o vetor estiver ordenado em ordem não decrescente; (ii) -1, se o vetor estiver ordenado em ordem não crescente; e (iii) 0 se o vetor não estiver ordenado.

14. Considere as bases numéricas: decimal, binária, octal e hexadecimal. Escreva funções para converter números em diferentes bases, isto é, suas funções devem receber um número em uma das bases e convertê-lo para a base desejada. Faça funções para todas as combinações possíveis.