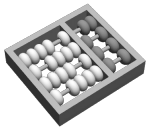

Universidade Estadual de Campinas
INSTITUTO DE COMPUTAÇÃO

MC 102 – Algoritmos e Programação de Computadores
Lista de Exercícios II

1 - Dadas duas seqüências, uma de *n* e a outra de *m* números inteiros entre 0 e 9, escrever um programa (utilizando funções) que calcule a seqüência de números inteiros entre 0 e 9 que represente a soma. Exemplo:

```
primeira seqüência:        3  4  5  1  8  0  5
segunda seqüência:               7  3  1  1  8
resultado:                 3  5  2  4  9  2  3
```

2 - Dadas duas seqüências de *n* e *m* valores inteiros, onde *n* <= *m*, escrever um programa (utilizando funções) que verifique quantas vezes a primeira seqüência ocorre na segunda. Exemplo:

```
primeira seqüência:  1 0 1
segunda seqüência:   1 1 0 1 0 1 0 0 1 1 0 1 0
                       -----           -----
                           -----
resultado: 3
```

3 - Escreva um procedimento para um programa que intercala os valores de dois vetores inteiros crescentes de mesmo tamanho em um terceiro vetor, em ordem crescente. Exemplo:

v1 = {1, 3, 5, 5, 7, 9, 10}

v2 = {2, 2, 4, 6, 8, 8, 10}

v3 = {1, 2, 2, 3, 4, 5, 5, 6, 7, 8, 8, 9, 10, 10}

4- Complete as lacunas ou marque com um traço as que não devem ser preenchidas.

O programa abaixo lê o tamanho dos catetos de um triângulo retângulo e imprime o tamanho da hipotenusa e a área do triângulo.

```
#include <stdio.h>
#include <math.h>

_____pitagoras(float __cat1, float __cat2, float __hip, float __area)
{
      ____hip = sqrt(____cat1 * ____cat1 + ____cat2 * ____cat2);
      ____area = ____cat1 * ____cat2/2.0;
}
```

```
int main(void)
{
	float c1, c2, h, a;

	printf("Insira o tamanho dos catetos do triangulo retangulo\n");
	scanf("%f %f", &c1, &c2); /* Lê o temanho dos catetos */

	calc_pitag(_________c1, ________c2, _______h, ________a);

	printf("A hipotenusa do triangulo retangulo eh: %.2f\n", h);
	printf("A area do triangulo retangulo eh: %.2f\n", a);

	return 0;
}
```

5 - Simule o seguinte código e diga quais os valores associados aos itens de (a) a (j). Utilize como entrada os díıgitos do seu RA, separados por espaço.

(a) *p1	(b) *p2	(c) *p3	(d) *p4	(e) *(vetor+5)
(f) vetor[7]	(g) *(p1 + 1)	(h) *p1 + 1	(i)vetor[8]	(j) *p2 + *p3

```
#include <stdio.h>

int main(void)
{
	int d1, d2, d3, d4, d5, d6;
	int vetor[10], *p1, *p2, *p3, *p4, i;

	scanf("%d %d %d %d %d %d", &d1, &d2, &d3, &d4, &d5, &d6);

	p1 = vetor;

	vetor[0] = d1 + d2;
	vetor[1] = d2 + d3;
	vetor[2] = d3 + d4;
	vetor[3] = d4 + d5;
	vetor[4] = d5 + d6;

	p2 = p1 + 1;
	p2++;
	p3 = p2 + 4;
	p4 = &vetor[9];

	*(vetor + 5) = 10;

	for (i = 0; i < 4; i++)
		*(p3 + i) = vetor[i];

	return 0;
}
```

6 - Considere o programa abaixo. Depois de executado, quais são os valores associados aos itens de (a) a (g). Suponha que os endereços das variáveis u e v são 1000 e 1004 respectivamente.

```
#include <stdio.h>
int main(void)
{
      int v, u;
      int *pv, *pu;
      v = 45;
      pv = &v;
      *pv = v + 1;
      u = *pv + 1;
      pu = &u;
      return 0;
}
```

(a) &v (b) pv (c) *pv (d) u
(e) &u (f) pu (g) *pu

7 - Considere o programa abaixo. Depois de executado, quais são os valores associados aos itens de (a) a (i). Suponha que os endereços das variáveis a, b e c são 1000, 1004 e 1008 respectivamente.

```
#include <stdio.h>
int main(void)
{
      float a, b;
      float c, *pa, *pb;
      a = 1;
      b = 3;
      pa = &a;
      *pa = 2 * a;
      pb = &b;
      c = 3 * (*pa + *pb);
      return 0;
}
```

(a) &a (b) &b (c) &c (d) pa (e) *pa
(f) &(*pa) (g) pb (h) &(*pb) (i) c

8 - Escreva um programa que leia um número inteiro n e um vetor de números reais com n valores. Além disso, o programa deve conter a seguinte função:

```
int estat(float vet[], int n, float *media, float *mediana, float *desv);
```

A função deverá receber como parâmetro o vetor com `n` valores e calcular as medidas solicitadas (através dos parâmetros `media`, `mediana` e `desvio`, passados por referência). Além disso, a função deverá retornar com o uso do comando `return` a quantidade de elementos do vetor no intervalo [`media` – `desvio` , `media` + `desvio`]. Todas as informações calculadas devem ser impressas na `main`, após a chamada da função.

9 - Faça um programa que contêm uma função recursiva que calcula o valor de *m* a partir de um vetor de números reais *V* com tamanho *N* e um valor λ . Assuma $0 < \lambda < 1$.

$$m = \sum_{t=0}^{N-1} \lambda^{(N-t)} V_t$$

10 - Faça um programa que contêm uma função recursiva que calcula a média aritmética dos elementos de um vetor de números reais *V* com tamanho *N*.

11 - Calcular a representação binária de um número consiste em realizar sucessivas divisões deste número por 2 e, quando seu resto for menor que 2, imprimir do último para o primeiro, todos os restos das divisões. Por exemplo:

6 / 2 = 3 (resto 0) => 3 / 2 = 1 (resto 1) => 1 / 2 = 0 (resto 1) 6 em binário = 110

Desenvolva a função recursiva para o cálculo da representação binária de um inteiro positivo *n* qualquer.

12 - Desconsidere o conhecimento da função *sqrt* na linguagem C. Uma forma de se obter a raiz quadrada de um número qualquer x seria através de busca binária. Assuma que a raiz quadrada de x está entre 0 e x (Se o número for negativo, retorne 0). Para sabermos se um palpite y é a raiz quadrada de x, basta testar se y*y é próximo o suficiente de x ou, em outras palavras, se o módulo da diferença entre eles está dentro de uma tolerância definida. Caso contrário, podemos restringir a busca entre 0 e y ou entre y e x. Escreva a função abaixo que implemente este algoritmo, considerando 10^{-6} como tolerância para o cálculo do resultado.

double raiz(double x);

13 - Crie um programa que dado uma string, coloque as letras dela em ordem crescente pelo algoritmo da seleção.

14 - Faça um programa que leia *n* nomes e ordene-os pelo tamanho utilizando o algoritmo da inserção. No final, o algoritmo deve mostrar todos os nomes ordenados.

15 - Faça um programa que leia *n* nomes e ordene-os pelo tamanho utilizando o algoritmo mergesort. No final, o algoritmo deve mostrar todos os nomes ordenados.

16 - Crie um programa que dado uma string, coloque as letras dela em ordem crescente pelo algoritmo quick-sort.

17 - As moléculas de DNA podem ser representadas por cadeias de caracteres que usam um alfabeto de 4 letras: A (Adenina), C (Citosina), T (Timina) e G (Guanina). Crie uma ***função*** e um programa que utilize essa função, que procure ocorrências de uma subcadeia de DNA dentro de uma outra cadeia de DNA. Você deverá procurar somente ocorrências diretas. Por exemplo, se:

cadeia = TCATATGCAAATAGCTGCATACCGA
subcadeia = CATA

Então a subcadeia ocorre na forma direta na posição 2 e na posição 18 da cadeia principal.

18 - A função *scanf* da biblioteca *stdio.h*, quando aplicada a floats, considera o "." como separador decimal, enquanto que no Brasil nós usamos ",".

Sua tarefa é criar uma scanf brasileira, a scanfBrasil, que deve ler somente floats escritos na forma "nn,nn" (com um número qualquer de dígitos) e guardá-los em uma variável float. Sua scanfBrasil deve funcionar de forma semelhante a scanf.

Além disso, sua scanfBrasil deve retornar 1 se tudo correu bem, e 0 em caso de erro (quando o valor dado não é nem float nem inteiro, por exemplo). No caso de erro, a scanfBrasil não deve armazenar nada na variável a ela passada. Utilize a main:

```
int main(void)
{
     double f;
     if (scanfBrasil(&f) != 0)
          printf("Erro!\n");
     else
          printf("Numero = %f\n",f);
     return 0;
}
```

Dica: Leia a entrada como uma string e verifique se ela contém somente números (inteiro), ou números com separador ',' (float).