

Lista de Exercícios

27) Erros

a) O seguinte programa promete simular uma jogada de um dado de 6 faces. Qual o defeito do programa?

```
int RolaDado (void) {  
    int r = rand ();  
    if (r < RAND_MAX/6) return 1;  
    else if (r < 2 * RAND_MAX/6) return 2;  
    else if (r < 3 * RAND_MAX/6) return 3;  
    else if (r < 4 * RAND_MAX/6) return 4;  
    else if (r < 5 * RAND_MAX/6) return 5;  
    else return 6;  
}
```

b) O seguinte programa promete simular uma jogada de um dado de 6 faces. Qual o defeito do programa?

```
int RolaDado (void) {  
    int r = rand ();  
    if (r < RAND_MAX/6) return 1;  
    else if (r < RAND_MAX/6 * 2) return 2;  
    else if (r < RAND_MAX/6 * 3) return 3;  
    else if (r < RAND_MAX/6 * 4) return 4;  
    else if (r < RAND_MAX/6 * 5) return 5;  
    else return 6;  
}
```

c) O seguinte programa promete simular uma jogada de uma moeda. Qual o defeito do programa?

```
char *RolaMoeda (void) {  
    int r;  
    r = rand () % 2;  
    if (r == 1) return "cara";  
    else return "coroa";  
}
```

46) Alocação de Memória

a) Escreva uma função que receba um byte `c` (que pode representar um [caractere ASCII](#), por exemplo) e transforme-o em uma [string](#), ou seja, devolva uma string de comprimento 1 tendo `c` como único elemento.

b) O que há de errado com o seguinte fragmento de código?

```
int *v;  
v = malloc (100 * sizeof (int));  
v[0] = 999;  
free (v+1);
```

c) A seguinte função promete devolver um vetor com os 3 primeiros números primos maiores que 1000. Onde está o erro?

```
int *primos (void) {  
    int v[3];  
    v[0] = 1009; v[1] = 1013; v[2] = 1019;  
    return v; }
```

d) Discuta, passo a passo, o efeito do seguinte fragmento de código:

```
int *p, *q;  
p = malloc (sizeof (int));  
*p = 123;  
q = malloc (sizeof (int));  
*q = *p;  
q = p;  
free (p);  
free (q); // má ideia...  
q = NULL; // boa ideia
```

e) Escreva um programa que leia um número inteiro positivo n seguido de n números inteiros e imprima esses n números em ordem invertida. Por exemplo, ao receber

5 22 33 44 55 88

o seu programa deve imprimir:

88 55 44 33 22

O programa não deve impor limitações sobre o valor de n.

f) Para uma dada string, contendo uma linha de texto, faça uma função em C que separa cada palavra do texto:

A função deve devolver o endereço de um vetor de apontadores criado dinamicamente com tamanho igual ao número de palavras do texto. Os apontadores desse vetor devem apontar para cópias das palavras do texto. Cada cópia deve ser armazenada em um vetor alocado dinamicamente, usando a menor quantidade de memória possível.

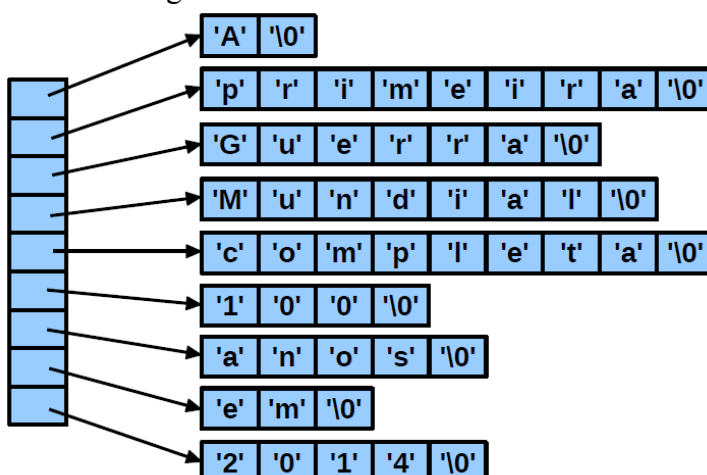
Use o protótipo abaixo:

```
char **VetorDePalavras(char texto[], int *npal);
```

tal que **npal** aponta para uma variável que irá guardar o número de palavras encontradas.

Exemplo:

Para o texto de entrada: "A primeira Guerra Mundial completa 100 anos em 2014", no final teremos os seguintes vetores alocados:



Exercícios Complementares:

<https://www.ime.usp.br/~macmulti/exercicios/extra/index.html>