Lista de exercícios: recursão

Implemente funções recursivas para fazer o que é pedido em cada exercício.

Exercício 1. Computar a soma dos primeiros n inteiros positivos.

```
int soma(int n);
```

Exercício 2. Computar a soma dos elementos de um vetor `v` com índices de 0 até $n-1$.

```
int soma(int n, int *v);
```

Exercício 3. Encontrar o valor do menor elemento de um vetor `v` com índices de 0 até $n-1$.

```
int min(int n, int *v);
```

Exercício 4. Decidir se uma palavra `p` é palíndroma.

```
int eh_palindroma(int n, char *p);
```

Exercício 5. Reverter a ordem das letras de uma palavra `p`.

```
void reverter(int n, char *p);
```

Exercício 6. Calcular 2^k.

```
long pot2(int k);
```

Exercício 7. Dado k, enumerar as potências de 2 até 2^k.

```
void enum_pot2(int k);
```

Exercício 8. Dados inteiros positivos n e b, imprimir n em base b.

```
void imprimir_em_base(int n, int b);
```

Exercício 9. Dado um inteiro positivo n, calcular a soma dos dígitos de n (em base 10).

```
int soma_dígitos(int n);
```

Exercício 10. Dado um vetor `v` com n elementos, imprimir todas as $n!$ permutações dos elementos de v.

```
void enum_permut(int n, int *v);
```

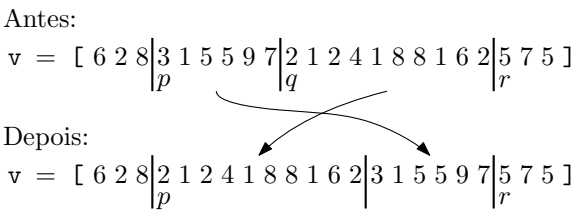
Se você preferir, a função `enum_permut` não precisa ser a função recursiva, mas você pode fazer outra função que será a função recursiva propriamente dita, de maneira que `enum_permut` chame essa outra função (seja um *wrapper* para o uso dessa outra função).

Exercício 11. Dado um vetor `v` com n elementos, possivelmente repetidos, imprimir todas as permutações distintas dos elementos de v.

```
void enum_permut(int n, int *v);
```

Mesmo caso do exercício anterior.

Exercício 12. Suponha que `v` seja um vetor de inteiros. Sejam $p < q < r$ índices válidos desse vetor. Considere o problema de trocar os elementos dos subvetores `v[`$p..(q-1)$`]` e `v[`$q..(r-1)$`]` de lugar, preservando a ordem em cada subvetor. Veja um exemplo do que isso significa na figura abaixo.



Descreva o código em C de um algoritmo que resolve o problema recursivamente usando espaço extra constante (que não dependa do tamanho do vetor de entrada). **Isto é, você não pode usar um vetor auxiliar!**

```
void troca_blocos(int p, int q, int r, int *v);
```