

Exercícios com *arrays* na linguagem C

Com esta lista de exercícios o aluno pratica a escrita de programas que envolvam a utilização e *arrays* unidimensionais (vectores) e bidimensionais .

1. Realize a função

```
void show_array_range( int vals[], int start, int end, bool growing )
```

que apresenta na consola os elementos do *array* vals presentes nos índices entre start e end, inclusivé (start deverá ser menor do que end). O parâmetro growing define a ordem de apresentação (de start para end, se growing for true, ou de end para start, caso o valor seja *false*).

2. Realize a função

```
void invertArray(int vals[], int size)
```

que inverte os elemento do *array* vals, isto é, o último troca com o primeiro, o penúltimo com o segundo, e assim sucessivamente.

3. Realize a função

```
void rotateRight(int vals[], int size)
```

Que roda de uma casa para a direita os elementos do *array* vals. O que está última posição (size -1) passa para a posição 0.

4. Considerando as seguintes definições:

```
#define LINES  25
#define COLS   25
typedef int  Matrix[LINES][COLS];
```

- a) Realize função void matrixCross(Matrix m) que preenche a matriz m de acordo com o padrão apresentado na Fig. 1 (exemplo com 9 x 9).

1	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	1	0
0	0	1	0	0	0	1	0	0
0	0	0	1	0	1	0	0	0
0	0	0	0	1	0	0	0	0
0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	1	0	0
0	1	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1

Fig. 1

- b) Realize a função `void matrixBorder(Matrix m)` que preenche a matriz `m` de acordo com o padrão apresentado na Fig. 2 (exemplo com 9 x 9).

1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1

Fig. 2

5. Seja a representação do triângulo de Pascal em matriz, mostrada na fig. A. A 1ª linha e a primeira coluna contêm 1s. O valor de cada célula a partir de 1,1 é dado por $M_{i,j} = M_{i,j-1} + M_{i-1,j}$.

1	1	1	1	1	1	1
1	2	3	4	5	6	
1	3	6	10	15		
1	4	10	20			
1	5	15				
1	6					
1						

Fig. 4

1	1	1	1	1	1	1
1	2	3	4	5	6	
1	3	5	10	15		
1	4	10	20			
1	5	15				
1	6					
1						

Fig. 5

Considerando o tipo: `typedef int Matrix[N][N];`

Realize a função `bool pascal_triangle(Matrix m)` que retorna `true` ou `false`, respectivamente, se a matriz `m` passada em parâmetro corresponde, ou não, à representação do triângulo de Pascal. Por exemplo, a matriz da fig. B não representa o triângulo de Pascal.