Ficha 3

Programação Imperativa

Vectores de inteiros

Pode codificar as soluções dos vários exercícios na página `https://codeboard.io/projects/224189`.

1. Diga, justificando, qual o output de cada um dos seguintes excertos de código C.

   (a) `https://tinyurl.com/4wcnjzpn`

   ```
   int main () {
       int x [15] = {1,  2, 3, 4, 5,
                     6,  7, 8, 9,10,
                     11,12,13,14,15};
       int *y, *z, i;
       y = x;
       z = x+3;
       for (i=0; i<5; i++) {
         printf ("%d %d %d\n",
                 x[i], *y, *z);
         y = y+1; z = z+2;
   }
   ```

   (b) `https://tinyurl.com/b5jzrh69`

   ```
   int main () {
     int i, j, *a, *b;

     i=3; j=5;
     a = b = 42;
     a = &i; b = &j;
     i++;
     j = i + *b;
     b = a;
     j = j + *b;
     printf ("%d\n", j);

     return 0;
   }
   ```

2. Defina uma função `void swapM (int *x, int *y)` que troca o valor de duas variáveis. Por exemplo, o código ao lado deverá imprimir no ecran `5 3`.

   ```
   int x = 3, y = 5;
   swapM (&x, &y);
   printf ("%d %d\n", x, y);
   ```

3. Defina uma função `void swap (int v[], int i, int j)` que troca o valor das posições `i` e `j` do vector `v`.

4. Defina uma função `int soma (int v[], int N)` que calcula a soma dos elementos de um vector `v` com `N` inteiros.

5. Defina uma função `void inverteArray (int v[], int N)` que inverte um array. Escreva duas versões, cada uma usando uma das funções das alíneas anteriores.

6. Defina uma função `int maximum (int v[], int N, int *m)` que coloca em `*m` o maior dos elementos do vector `v`.

   A função deverá retornar `0` sse tal for possível (i.e., quando `N>0`).

7. Defina uma função `void quadrados (int q[], int N)` que preenche o vector `q` com os quadrados dos primeiros `N` números naturais (`{0,1,4,9,...}`.

   Note que, uma vez que $(a+1)^2 = a^2 + (2*a+1)$ esta função não precisa de calcular o quadrado de nenhum dos números explicitamente.

8.

O triângulo de Pascal é uma forma de calcular os coeficientes da expansão do binómio de Newton. Ao lado relembramos as 5 primeiras linhas. Note que a linha n do triângulo tem n elementos e que a linha $n+1$ pode ser obtida a partir da linha n usando o seguinte processo:

```
                1
              1   1
            1   2   1
          1   3   3   1
        1   4   6   4   1
      1   5  10  10   5   1
```

- acrescenta-se um 1 no final, i.e., coloca-se `1` na posição `n`.
- para todos os elementos (desde `n-1` até `1`, por esta ordem) substitui-se o elemento nessa posição pela sua soma com o que está na posição anterior.

(a) Defina uma função `void pascal (int v[], int N)` que preenche o vector `v` com a `N`-ésima linha do triângulo de Pascal.

(b) Adapte a função que definiu atrás para, em vez de preencher um array com a linha `N` do triângulo, escreva no ecran as N primeiras linhas do triângulo.