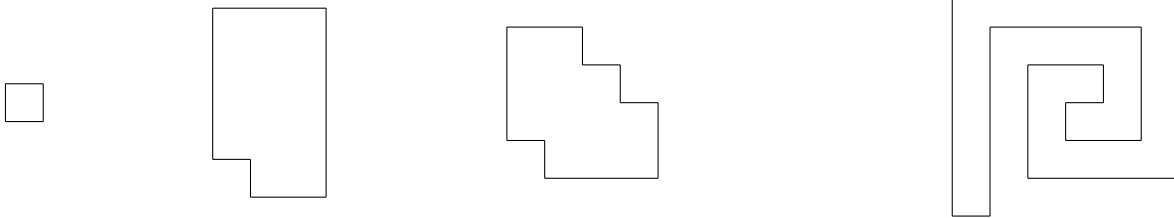


Folha 6

1. Implementar as funções seguintes:

- a) `void imprimeLinha(int mat[][NMAX], int k, int n)` para imprimir os elementos da matriz que estão na linha `k`, nas `n` primeiras colunas. Devem ser escritos numa mesma linha.
- b) `int maximo(int mat[][NMAX], int m, int n)` que retorna o máximo dos elementos da matriz, supondo que tem `m` linhas e `n` colunas e apenas elementos positivos.

2. Um polígono ortogonal, como os desenhados na figura, pode ser definido pela sequência dos seus vértices, dada no sentido horário. Admita que as coordenadas dos vértices são inteiros não negativos.



a) Escreva uma função `void imprimePol(int v[][2], int n)` que recebe em `n` o número de vértices do polígono e na matriz dada por `v` as coordenadas dos seus vértices, ordenados no sentido horário, e produz um resultado como os seguintes.

```
\documentclass{article}
\setlength{\unitlength}{0,5cm}
\begin{document}
\begin{picture}(3,5)
\put(1,1){\line(-1,0){1}}
\put(0,1){\line(0,1){4}}
\put(0,5){\line(1,0){3}}
\put(3,5){\line(0,-1){5}}
\put(3,0){\line(-1,0){2}}
\put(1,0){\line(0,1){1}}
\end{picture}
\end{document}
```

```
\documentclass{article}
\setlength{\unitlength}{0,5cm}
\begin{document}
\begin{picture}(6,6)
\put(0,6){\line(1,0){6}}
\put(6,6){\line(0,-1){5}}
\put(6,1){\line(-1,0){4}}
\put(2,1){\line(0,1){3}}
\put(2,4){\line(1,0){2}}
\put(4,4){\line(0,-1){1}}
\put(4,3){\line(-1,0){1}}
\put(3,3){\line(0,-1){1}}
\put(3,2){\line(1,0){2}}
\put(5,2){\line(0,1){3}}
\put(5,5){\line(-1,0){4}}
\put(1,5){\line(0,-1){5}}
\put(1,0){\line(-1,0){1}}
\put(0,0){\line(0,1){6}}
\end{picture}
\end{document}
```

(Guarde cada um dos dois resultados num ficheiro, por exemplo, `poly.tex`, e execute na *shell* o comando `pdflatex poly.tex`, o qual produz um ficheiro `poly.pdf`.)

À esquerda, $n = 6$ e $v[][2] = \{(1,1), (0,1), (0,5), (3,5), (3,0), (1,0)\}$. À direita, $n = 12$ e $v[][2] = \{(0,6), (6,6), (6,1), (2,1), (2,4), (4,4), (4,3), (3,3), \dots, (1,5), (1,0), (0,0)\}$.

Os dois valores indicados em `\begin{picture}(Mx,My)` correspondem aos máximos das coordenadas. Em cada linha, `\put(i,j)` identifica o vértice em que está e `\line(dx,dy){k}` define um segmento de recta com extremos nos vértices (i,j) e $(i,j)+k(dx,dy)$, podendo dx e dy tomar apenas os valores 1, -1 e 0. Os segmentos são horizontais ou verticais. Notar que, o par (dx,dy) define a direcção em que se desloca para chegar de (i,j) ao vértice seguinte e k dá o número de posições que teria de se deslocar. As coordenadas dos vértices são não negativas.

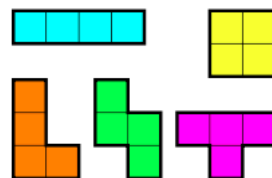
NB: Para imprimir `\line` tem de escrever `printf("\\line");` e, analogamente, sempre que for necessário escrever o carácter `\` deve escrever `\\` no formato.

b) Escreva um procedimento `void dilata(int v[][2],int n,int p,int q)` para transformar as coordenadas dos vértices. A transformação corresponde a aumentar de uma unidade a abcissa x se $x \geq p$ e a ordenada y se $y \geq q$. Verificar o efeito, usando a função definida na alínea anterior.

3. Escreva uma função `int repeteLinha(int mat[][NMAX],int m,int n,int i)` para duplicar a linha i da matriz, deslocando as linhas a partir de $i+1$ para baixo de forma a criar espaço livre para a operação. Suponha que a função só será chamada se $m+1$ não exceder o número de linhas com que `mat` foi definida.

Analogamente, defina `int repeteColuna(int mat[][NMAX],int m,int n,int j)` para duplicar a coluna j da matriz, deslocando as colunas a partir de $j+1$ para a direita de forma a criar espaço livre para a operação. Suponha que a função só será chamada se $n+1$ não exceder `NMAX`.

4. Suponha que um **poliominó** é representado por uma matriz de 0's e 1's (*bitmap*), com m linhas e n colunas, em que 0 designa posição livre e 1 posição ocupada. Cada elemento da matriz é do tipo `char` (mas, as posições guardam os valores 0 e 1, e não '0' e '1').



a) Escreva a função `void impr(char p[][NMAX],int m,int n,char c)` para imprimir o poliominó dado por `p`, usando o carácter `c` para identificar posições preenchidas e espaço para posições livres. Por exemplo, se `c = 'x'`, poderia ter os poliominós desenhados à direita.

```

xx      xxxxxx
 x      x      x
        x  xx  x
        x  x   x
        x  xxxx
        x

```

b) Pretendemos aplicar a operação de *dilatação* de uma célula (i,j) , com preservação da estrutura do poliominó, o que obriga a duplicação da linha i e da coluna j . Implemente a operação por aplicação das funções definidas no problema anterior. À direita, o resultado, para $(i,j) = (0,1)$ e $(i,j) = (4,1)$ respetivamente.

```

xxx      xxxxxxxx
xxx      x        x
        xx      x  xx  x
        x      x   x  x
        x  xxxx
        x  xxxx
        x

```

c) Admita que a matriz que define o poliominó é quadrada (i.e., $m=n$) e n é ímpar. Escreva uma função `void rodar(char p[][NMAX],int n,int graus)` para efetuar uma rotação do poliominó. O centro da rotação é a célula $(n/2,n/2)$ e a amplitude é dada por `graus`, sendo um múltiplo de 90. A rotação será efetuada no sentido horário. Comece por estudar o problema matematicamente.