

1. (Adaptado da OBI 2005)

Minhocas são muito importantes para a agricultura e como insumo para produção de ração animal. A Organização para Bioengenharia de Minhocas (OBM) é uma entidade não governamental que promove o aumento da produção, utilização e exportação de minhocas.

Uma das atividades promovidas pela OBM é a manutenção de uma fazenda experimental para pesquisa de novas tecnologias de criação de minhocas. Na fazenda, a área destinada às pesquisas é de formato retangular, dividida em células quadrangulares de mesmo tamanho. As células são utilizadas para testar os efeitos, na produção de minhocas, de variações de espécies de minhocas, tipos de terra, de adubo, de tratamento, etc. Os pesquisadores da OBM mantêm um acompanhamento constante do desenvolvimento das minhocas em cada célula, e têm uma estimativa extremamente precisa da produtividade em cada uma das células. A tabela abaixo mostra um mapa da fazenda, mostrando a produtividade estimada de cada uma das células.

81	28	240	10
40	10	100	240
20	180	110	35

Um pesquisador da OBM inventou e construiu uma máquina colhedeira de minhocas, e quer testá-la na fazenda. A máquina tem a largura de uma célula, e em uma passada pelo terreno de uma célula colhe todas as minhocas dessa célula, separando-as, limpando-as e empacotando-as. Ou seja, a máquina eliminará uma das etapas mais intensivas de mão de obra no processo de produção de minhocas. A máquina, porém, ainda está em desenvolvimento e tem uma restrição: não faz curvas, podendo movimentar-se somente em linha reta.

Decidiu-se então que seria efetuado um teste com a máquina, de forma a colher o maior número possível de minhocas em uma única passada, em linha reta, de lado a lado do campo de minhocas. Ou seja, a máquina deve colher todas as minhocas de uma “coluna” ou de uma “linha” de células do campo de minhocas (a linha ou coluna cuja soma das produtividades esperadas das células é a maior possível).

Escreva um programa que, fornecido o mapa do campo de minhocas, descrevendo a produtividade estimada em cada célula, calcule o número esperado total de minhocas a serem colhidas pela máquina durante o teste, conforme descrito acima.

Entrada:

Para simplificar o código, as entradas serão digitadas diretamente no código-fonte. Use as macros `N_LINHAS` e `N_COLUNAS` para determinar o tamanho do campo. Já o campo pode ser informado diretamente como uma matriz (veja o exemplo na próxima página).

Saída:

O programa deve mostrar um inteiro, indicando o número esperado total de minhocas a serem colhidas pela máquina durante o teste.

Exemplos (pode copiar as entradas para o seu código!):

```
#define N_LINHAS 3
#define N_COLUNAS 4

int campo [N_LINHAS][N_COLUNAS] = {{81, 28, 240, 10},
                                     {40, 10, 100, 240},
                                     {20, 180, 110, 35}};
```

Saída esperada: 450

```
#define N_LINHAS 4
#define N_COLUNAS 1

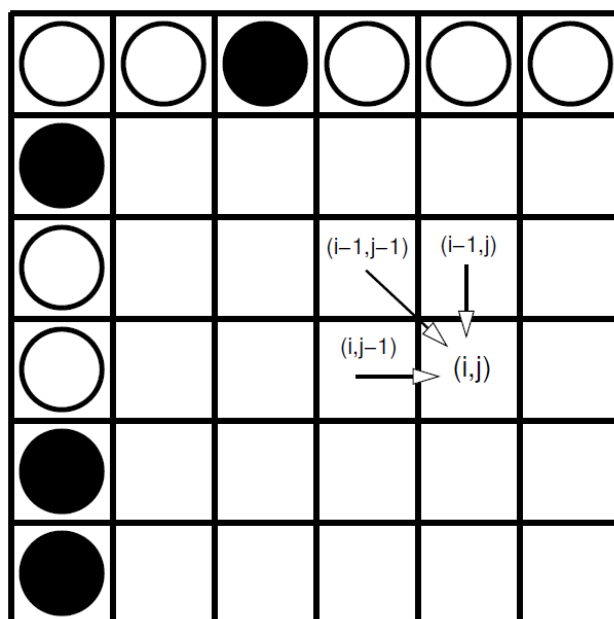
int campo [N_LINHAS][N_COLUNAS] = {{100},
                                     {110},
                                     {0},
                                     {100}};
```

Saída esperada: 310

(**) 2. Escreva um programa que lê N strings de até 128 caracteres e depois mostra na tela as strings em ordem inversa à fornecida (i.e. o programa mostra a última string, depois a penúltima, e assim por diante, e no final mostra a primeira string digitada).

3. (Adaptado da OBI 2017)

Flavinho não se cansa de bolar joguinhos para passar o tempo. Ele diz que é uma boa forma de treinar a memória e a capacidade de resolver problemas. Dessa vez ele inventou uma forma de preencher um tabuleiro de N linhas e N colunas com pedras brancas e pretas. Inicialmente ele coloca, aleatoriamente, pedras brancas e pretas em todas as células da primeira coluna e da primeira linha. A figura abaixo dá um exemplo de tabuleiro com $N = 6$. Ele chama essas pedras iniciais de sementes. Uma vez colocadas as sementes, as demais células do tabuleiro serão preenchidas com uma pedra branca ou preta.



Considere a célula na posição (i,j) , para $i > 0$ e $j > 0$. Para saber a cor da pedra nessa célula, Flavinho precisa saber a cor das pedras nas três células $\{(i,j-1), (i-1,j-1), (i-1,j)\}$. A figura também ilustra quais células são usadas para determinar a cor da pedra na célula (i,j) . Se houver mais pedras brancas do que pretas nessas três células, a cor da pedra na célula (i,j) será preta. Se houver mais pedras pretas do que brancas, a cor será branca.

Note que, por essa definição, a primeira célula a ser preenchida será a $(1,1)$, pois será a única vazia para a qual já saberemos a cor das três pedras necessárias. No exemplo da figura, a pedra na célula $(1,1)$ será da cor preta, pois há duas brancas e uma preta entre as células $\{(1,0),(0,0),(0,1)\}$.

Neste problema, dado N e as cores das sementes, seu programa deve preencher o restante do tabuleiro. Para simplificar o código, as entradas serão digitadas diretamente no código-fonte. Use a macro `N` para determinar o tamanho do tabuleiro. Já o campo pode ser informado diretamente como uma matriz. Na representação interna, o valor 0 será usado para identificar uma pedra preta, 1 para uma pedra branca, e 9 para uma posição cujo conteúdo é desconhecido.

Exemplos (pode copiar as entradas para o seu código!):

```
#define N 2
int tabuleiro [N][N] = {{0,1},
                        {1,9}};
```

Saída esperada:

```
01
10
```

```
#define N 6
int tabuleiro [N][N] = {{0,0,1,0,0,0},
                        {1,9,9,9,9,9},
                        {0,9,9,9,9,9},
                        {0,9,9,9,9,9},
                        {1,9,9,9,9,9},
                        {1,9,9,9,9,9}};
```

Saída esperada:

```
001000
110111
001000
010111
101000
101011
```

(***) 4. Escreva um programa que gera duas matrizes A e B , respectivamente com $M \times N$ e $N \times M$ posições, e ambas contendo valores aleatórios. O programa deve mostrar o resultado da multiplicação $A \cdot B$. Lembre-se que a multiplicação de matrizes não é simplesmente a multiplicação elemento a elemento das matrizes. Por exemplo, suponha que $M=2$ e $N=3$. As matrizes ficam sendo:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \quad B = \begin{bmatrix} g & h \\ i & j \\ k & l \end{bmatrix}$$

A matriz resultante da multiplicação de A por B é então dada por:

$$\begin{bmatrix} ag+bi+ck & ah+bj+cl \\ dg+ei+fk & dh+ej+fl \end{bmatrix}$$

(*) 5. O Sudoku é um quebra-cabeça numérico que tem como objetivo preencher com valores de 1 a 9 um conjunto de 9×9 células em um grid, com base em alguns números iniciais e algumas regras: não podem existir números repetidos em uma mesma linha, nem em uma mesma coluna. Além disso, as células são divididas em 9 subconjuntos de 3×3 células, e não podem existir números repetidos em um mesmo subconjunto. A figura a seguir mostra um quebra-cabeça resolvido:

9	5	4	8	1	6	3	7	2
7	8	6	2	5	3	1	4	9
1	2	3	7	9	4	6	5	8
3	1	8	9	7	2	4	6	5
2	7	9	4	6	5	8	1	3
4	6	5	3	8	1	9	2	7
8	4	7	1	2	9	5	3	6
5	3	2	6	4	8	7	9	1
6	9	1	5	3	7	2	8	4

Escreva um programa que recebe uma solução de um quebra-cabeça como uma matriz de 9×9 inteiros. O programa deve dizer se a matriz contém uma solução válida de acordo com as regras do Sudoku. Para evitar o trabalho de ter que digitar os números a cada execução do programa, inicialize a matriz no próprio código, como no exemplo abaixo.

```
int resposta_sudoku [9][9] =
    {{9, 5, 4, 8, 1, 6, 3, 7, 2},
     {7, 8, 6, 2, 5, 3, 1, 4, 9},
     {1, 2, 3, 7, 9, 4, 6, 5, 8},
     {3, 1, 8, 9, 7, 2, 4, 6, 5},
     {2, 7, 9, 4, 6, 5, 8, 1, 3},
     {4, 6, 5, 3, 8, 1, 9, 2, 7},
     {8, 4, 7, 1, 2, 9, 5, 3, 6},
     {5, 3, 2, 6, 4, 8, 7, 9, 1},
     {6, 9, 1, 5, 3, 7, 2, 8, 4}};
```

6. (Adaptado de <https://neps.academy/br/exercise/483>)

*"Caneta azul, azul caneta
Caneta azul tá marcada com minhas letra
Caneta azul, azul caneta
Caneta azul tá marcada com minhas letra"*

Esses são alguns versos de uma música que virou meme impulsionado com a divulgação nas redes sociais de famosos. Simaria (da dupla com Simone) e Tirulipa, entre outras celebridades, postaram vídeos cantando a música. Em shows recentes, Wesley Safadão, Pablo do Arrocha e Alok cantaram trechos de "Caneta Azul".

Manoel, compositor da música, em seu relato diz: "Eu ia para o colégio e perdi a caneta azul. Lá no colégio eu perdia uma, perdia outra, aí ninguém me deu a caneta que tava meu registro nela. Eu não achei e, no outro dia, fiz a música". Manoel garante ainda que a tal caneta ainda não apareceu. 😊😊😊

O seu papel será construir um programa que ajude o Manoel a encontrar a sua "Caneta Azul". É dada uma matriz contendo somente letras maiúsculas. O programa deve procurar pela sequência de letras CANETA AZUL (sem o espaço!) e mostrar na tela a posição da letra C, caso a sequência seja encontrada, ou uma mensagem de erro caso contrário. A sequência aparece somente na horizontal, lida da esquerda para a direita. Como Manoel só perdeu uma caneta, a execução do programa deve parar assim que a caneta for encontrada!

Dica: Tente pensar em uma solução genérica, que funcione para qualquer sequência.

Exemplos (pode copiar as entradas para o seu código!):

```
char matriz [N_LINHAS][N_COLUNAS] =  
    {"CANETA AZULEO",  
     "RHHEEIOYHMBE",  
     "RNRYSENTOKWS",  
     "IAIMITCGWVSR",  
     "NHRNAURHEABK",  
     "RTARUHETTNAO",  
     "LMDCPNHMSATE",  
     "TSEGNNSDLSHA",  
     "OTWYCNLIHHRF",  
     "IDRENAOBE LRC",  
     "LTYETOTIFCEA",  
     "OFKAARATWANP"};
```

Saída esperada: 0 0

```
char matriz [N_LINHAS][N_COLUNAS] =  
    {"RHHEEIOYHMBE",  
     "RNRYSENTOKWS",  
     "IAIMITCGWVSR",  
     "NHRNAURHEABK",  
     "RTARUHETTNAO",  
     "AACANETA AZUL",  
     "LMDCPNHMSATE",  
     "TSEGNNSDLSHA",  
     "OTWYCNLIHHRF",  
     "IDRENAOBE LRC",  
     "LTYETOTIFCEA",  
     "OFKAARATWANP"};
```

Saída esperada: 5 2

(****) (OPCIONAL!) 7. Faça outra versão do programa da questão anterior, mas agora a sequência pode aparecer na horizontal ou na vertical, sendo lida da direita para a esquerda, da esquerda para direita, de cima para baixo ou de baixo para cima.

Exemplos:

```
char matriz [N_LINHAS][N_COLUNAS] =
    {"RHHEEIOYHMBE",
     "RCRYSENTOKWS",
     "IAIMITCGWVSR",
     "NNRNAURHEABK",
     "LUZAATENACAC",
     "ATXGADSFWERS",
     "LADCPNHMSATE",
     "TAEGNNSDLSHA",
     "OZWYCNLIHHRF",
     "IURENAOBELRC",
     "LRYETOTIFCEA",
     "XCANETA AZULA"};
```

Saída esperada: 4 9

```
char matriz [N_LINHAS][N_COLUNAS] =
    {"RHHEEIOYHMBE",
     "RCRYSENTOKWS",
     "IAIMITCGWVSR",
     "NNRNAURHEABK",
     "REARUHETTNAO",
     "ATXGADSFWERS",
     "LADCPNHMSATE",
     "TAEGNNSDLSHA",
     "OZWYCNLIHHRF",
     "IURENAOBELRC",
     "LLYETOTIFCEA",
     "XCANETA AZULA"};
```

Saída esperada: 1 1

```
char matriz [N_LINHAS][N_COLUNAS] =
    {"RHHEEIOYHMBE",
     "RCRYSENTOKWS",
     "IAILITCGWVSR",
     "NNRUAURHEABK",
     "REAZUHETTNAO",
     "ATXAADSFWERS",
     "LADAPNHMSATE",
     "TQETNNSDLSHA",
     "OQWECNLIHHRF",
     "IFRNNAOBELRC",
     "LEYATOTIFCEA",
     "XFACETA AZULA"};
```

Saída esperada: 11 3