

Assim, gradualmente a matriz terá marcas que ajudam a identificar novas marcas. Repetindo este processo, a matriz terá mais e mais marcas até que finalmente o quebra-cabeça estará completamente resolvido. Mas tome cuidado: é possível que apenas uma marca colocada incorretamente leve a um *deadlock*.

O Trabalho Prático

Você deve implementar um programa que lê de um arquivo os seguintes dados: (i) os tamanhos $n \times m$ da matriz; (ii) os cabeçalhos do *Nonograma* (primeiro das linhas e em seguida das colunas); e (iii) a matriz com o que está preenchido até o momento, representando o jogo.

Seguem os arquivos que descrevem as duas matrizes apresentadas na Figura 1:

Nonograma em branco		Nonograma preenchido	
1	6 7	1	6 7
2		2	
3	1 3	3	1 3
4	2 1 1	4	2 1 1
5	1 6	5	1 6
6	2 5 1	6	2 5 1
7	1 6	7	1 6
8	1 3	8	1 3
9		9	
10	1 4	10	1 4
11	2 1 4	11	2 1 4
12	2 1 4	12	2 1 4
13	2 1 4	13	2 1 4
14	1 4	14	1 4
15	2 1 1	15	2 1 1
16	1 1	16	1 1
17		17	
18	18	- x x x - - -
19	19	x - - - x - -
20	20	x x x x x x -
21	21	x x x x x - x
22	22	x x x x x x -
23	23	- x x x - - -

Note que cada arquivo pode ser dividido em quatro partes:

1. **Dimensão da matriz:** 6 linhas por 7 colunas.
2. **Cabeçalho das linhas:** cada linha apresenta a quantidade de números em uma linha seguido dos números.
3. **Cabeçalho das colunas:** cada linha representa uma coluna, e apresenta a quantidade de números seguido dos números.
4. **A matriz:** note que o caractere '.' representa células em branco, 'x' representa células coloridas e '-' representa células marcadas com um traço (ou seja, células que tem que ficar em branco). Notem que no arquivo da direita o usuário marcou as células em branco com '-', o que é opcional.

O arquivo contendo a descrição da matriz (tal como nos exemplos acima) deverá ser passado como um argumento na execução do programa.

O programa, logo após ser executado, deverá imprimir o *Nonograma*. Em seguida, o usuário será convidado a **digitar um comando** (ver possíveis comandos na tabela a seguir).

Comando	Argumento	Resultado
x	AC	Preenche a célula da linha A e coluna C com um 'x'.
-	EC	Preenche a célula da linha E e coluna C com um '-'.
.	BA	Limpa a célula da linha B e coluna A.
resolver		Resolve o <i>Nonograma</i> e marca automaticamente as células com 'x' ou '-'.
salvar	out.txt	Salva o <i>Nonograma</i> tal como está no momento no arquivo "out.txt".
sair		Encerra o programa (sem salvar as últimas alterações).

Após a execução de um comando (exceto **salvar** e **sair**), seu programa deve imprimir o *Nonograma*. Segue exemplo de execução do programa (dados digitados pelo usuário estão destacados em **azul**):

```

1  ./nonograma matriz-1.txt
2
3  Bem vindo ao Nonograma!
4
5      1 1 1 1
6      4 4 4 4 4 1 1
7      A B C D E F G
8  3 A . . . . .
9  1 1 B . . . . .
10 6 C . . . . .
11 5 1 D . . . . .
12 6 E . . . . .
13 3 F . . . . .
14
15 Digite um comando: x AB
16
17      1 1 1 1
18      4 4 4 4 4 1 1
19      A B C D E F G
20 3 A . x . . . . .
21 1 1 B . . . . .
22 6 C . . . . .
23 5 1 D . . . . .
24 6 E . . . . .
25 3 F . . . . .
26
27 Digite um comando: - BB
28
29      1 1 1 1
30      4 4 4 4 4 1 1
31      A B C D E F G
32 3 A . x . . . . .
33 1 1 B . - . . . . .
34 6 C . . . . .
35 5 1 D . . . . .
36 6 E . . . . .
37 3 F . . . . .
38
39 Digite um comando: x CB
40
41      1 1 1 1
42      4 4 4 4 4 1 1
43      A B C D E F G
44 3 A . x . . . . .
45 1 1 B . - . . . . .
46 6 C . x . . . . .
47 5 1 D . . . . .
48 6 E . . . . .
49 3 F . . . . .

```

```

50
51 Digite um comando: salvar parcial.txt
52 Nonograma salvo com sucesso!
53
54 Digite um comando: x DB
55
56      1 1 1   1
57      4 4 4 4 4 1 1
58      A B C D E F G
59 3 A . x . . . . .
60 1 1 B . - . . . . .
61 6 C . x . . . . .
62 5 1 D . x . . . . .
63 6 E . . . . . . .
64 3 F . . . . . . .
65
66 Digite um comando: salvar nova-parcial.txt
67 Nonograma salvo com sucesso!
68
69 Digite um comando: resolver
70
71      1 1 1   1
72      4 4 4 4 4 1 1
73      A B C D E F G
74 3 A - x x x - - -
75 1 1 B x - - - x - -
76 6 C x x x x x x -
77 5 1 D x x x x x - x
78 6 E x x x x x x -
79 3 F - x x x - - -
80
81 Digite um comando: sair

```

Importante: seu programa deve avisar quando o usuário concluir o jogo com sucesso. O usuário também deve ser alertado com uma mensagem de erro caso faça uma alocação que infrinja as regras do jogo.

Instruções

- O problema deve ser resolvido por meio de um programa em C ou C++.
- Não serão aceitos trabalhos que caracterizem cópia (mesma estrutura e algumas pequenas modificações) de outro.
- Após a entrega dos trabalhos serão marcadas entrevistas com cada um dos alunos para apresentação dos mesmos para um dos professores da disciplina.

Entrega

- A entrega do código-fonte será feita pelo Moodle até o dia **01 de Julho às 23:30**.
- Também deverá ser entregue um breve relatório (impresso e em mãos) sobre o trabalho contendo:
 - Descrição do problema tratado (não copiar este enunciado).
 - Relato das dificuldades encontradas durante a realização do trabalho e soluções encontradas.
 - Referências de sites e outros materiais utilizados para confecção de trabalhos, incluindo consultas a colegas (especificar quais).
- Atenção: o código-fonte do programa não deve ser incluído no relatório.

Avaliação

- Funcionamento adequado do programa.
- Atendimento ao enunciado do trabalho.
- Clareza do código (que deve ser devidamente comentado e indentado).
- Utilização de funções.
- Adequação da estrutura do programa (variáveis e comandos utilizados).
- Apresentação do trabalho e relatório.
- Compilação (códigos que não compilem serão zerados, e *warnings* diminuirão a nota).

Desafio!!!

- O comando **resolver** não é trivial de ser implementado; faça o melhor que puder!