

Universidade Federal de Ouro Preto – UFOP Instituto de Ciências Exatas e Biológicas – ICEB Departamento de Computação – DECOM

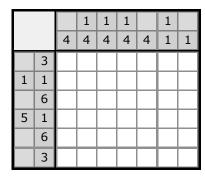
Disciplina: BCC201 - Introdução a Programação

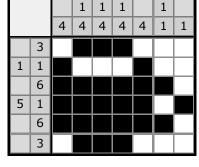
Professores: Túlio A. M. Toffolo e Puca Huachi V. Penna

## Trabalho Prático: **Nonograma**

O *Nonograma*, mais conhecido no Brasil pela alcunha *Logic Pix*, é um jogo de lógica (ou quebracabeça) japonês no qual uma figura é codificada em números. O objetivo é decodificar os números restaurando assim a figura.

O jogo consiste de uma matriz  $n \times m$  em que o cabeçalho contém os números que codificam uma figura. Estes números, mostrados à esquerda e acima da matriz, descrevem os grupos de quadrados pintados (que seguem sequencialmente, sem espaço) horizontal e verticalmente. A ordem desses números descreve a ordem de localização desses grupos de quadrados pintados, mas não se sabe onde cada grupo começa e termina. Na verdade, o desafio é justamente definir a localização dos grupos! Cada número indica um grupo separado do tamanho dado. Por exemplo, o número 5 indica um grupo de cinco quadrados pintados em sequência, enquanto o número 1 indica apenas um quadrado pintado. A figura a seguir apresenta um exemplo de Nonograma antes e após sua resolução:





(a) Matriz inicial

(b) Nonograma resolvido

Figura 1: Exemplo de Nonograma

### Como resolver um Nanograma?

Os passos de solução apresentados a seguir foram retirados do site http://www.nonograms.org/instructions.

Para resolver um *Nanograma*, deve-se olhar para cada linha/coluna separadamente, sempre movendo-se para as próximas colunas e linhas. Ao fazer isso, o processo de solução em cada linha/coluna consiste em:

- 1. Definir os quadrados que com certeza serão pintados.
- 2. Definir os quadrados em que é impossível ter quadrados pintados tais quadrados podem ser marcados, por exemplo, com um traço.
- 3. Definir os números que indicam grupos já marcados, e riscá-los.

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
     6 7
                                                                            6 7
 1
                                                                       1
     1 3
                                                                            1 3
                                                                            2 1 1
    2 1 1
 4
                                                                            1 6
    1 6
    2 5 1
                                                                            2 5 1
    1 6
                                                                            1 6
 8
    1 3
                                                                            1 3
    1 4
                                                                            1 4
10
    2 1 4
                                                                            2 1 4
11
     2 1 4
                                                                            2 1 4
12
13
14
    2 1 1
                                                                            2 1 1
15
16
17
                                                                      17
                                                                      18
18
                                                                      19
19
20
                                                                      20
21
                                                                            \mathbf{x} \mathbf{x} \mathbf{x} \mathbf{x} \mathbf{x}
22
                                                                            - 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 Nonograma e marca automaticamente as células com 'x' ou '-'.
salvar	out.txt	Salva o Nonograma 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**):

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

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

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!