

# Trabalho Prático 1 — Batalha Naval

## 1 Introdução

Batalha Naval é um jogo de tabuleiro para dois jogadores que consiste em cada um posicionar navios e, alternadamente, atacar uma coordenada  $(x, y)$ , tentando acertar a posição dos navios do adversário. Ganha o jogador que conseguir afundar a frota completa do outro.

## 2 Premissa

Na Terra da Computação está acontecendo uma grande guerra entre os Algoritmistas e os Estruturistas. Você foi recrutado pelo almirante das forças algoritmistas, Luisal Goritmos, para auxiliar no combate marítimo, sendo responsável pelo gerenciamento dos recursos computacionais da operação. Ao final de cada batalha, você deve analisar os dados dos radares e averiguar o resultado dos confrontos. Sendo assim, sua função é fazer um programa que consiga determinar quem venceu uma dada batalha, ou se houve um empate. Um empate ocorre quando ambos os lados terminaram o confronto com **pelo menos parte** de um navio ainda operante, ou seja, quando não houve destruição total de nenhum lado. Os dados dos radares foram pré-processados e transformados em um documento de formato mais amigável; ao invés de trabalhar com mapas, radares, e registros de caixas preta das embarcações afundadas, toda a informação de posicionamento foi codificada em uma matriz, e os ataques de cada frota são representados pelas posições nessa matriz acertada por eles.

O planeta PNP, onde se encontra a Terra da Computação, é um lugar meio estranho: ao invés de ser esférico como a nossa Terra, ele é toroidal, ou seja, ele tem o mesmo formato de um anel. Sendo assim, batalhas navais em PNP são muito mais interessantes: as bordas da matriz estão conectadas, de modo que uma célula na borda direita está conectada a uma célula na esquerda (veja a Figura 1), assim como uma célula na borda inferior está conectada a uma célula na superior (veja a Figura 2). Nesse sentido, quando um navio se move além de uma borda da matriz, ele “volta” para o outro lado da matriz.

Um outro fenômeno bem interessante dos confrontos em PNP é a existência dos chamados ataques especiais. Após dois ataques consecutivos bem sucedidos (ou seja, que acertaram uma embarcação inimiga), podem ser convocados

**Navio 1 (Azul):**

- Tamanho do navio = 5
- Orientação = Horizontal (**Sempre para direita**)
- Coordenada (Linha Coluna) = 3 9

Corresponde a uma entrada: 5 h 3 9

**Navio 2 (Vermelho):**

- Tamanho do navio = 3
- Orientação = Horizontal (**Sempre para direita**)
- Coordenada (Linha Coluna) = 6 7

Corresponde a uma entrada: 3 h 6 7

**Navio 3 (Verde):**

- Tamanho do navio = 4
- Orientação = Vertical (**Sempre para baixo**)
- Coordenada (Linha Coluna) = 5 2

Corresponde a uma entrada: 4 v 5 2



Figura 1: Exemplo de navio que dá a volta horizontalmente na matriz.

**Navio 1 (Azul):**

- Tamanho do navio = 5
- Orientação = Vertical (**Sempre para baixo**)
- Coordenada (Linha Coluna) = 9 9

Corresponde a uma entrada: 5 v 9 9

**Navio 2 (Vermelho):**

- Tamanho do navio = 4
- Orientação = Vertical (**Sempre para baixo**)
- Coordenada (Linha Coluna) = 6 2

Corresponde a uma entrada: 4 v 6 2

**Navio 3 (Verde):**

- Tamanho do navio = 5
- Orientação = Horizontal (**Sempre para direita**)
- Coordenada (Linha Coluna) = 3 2

Corresponde a uma entrada: 5 h 3 2

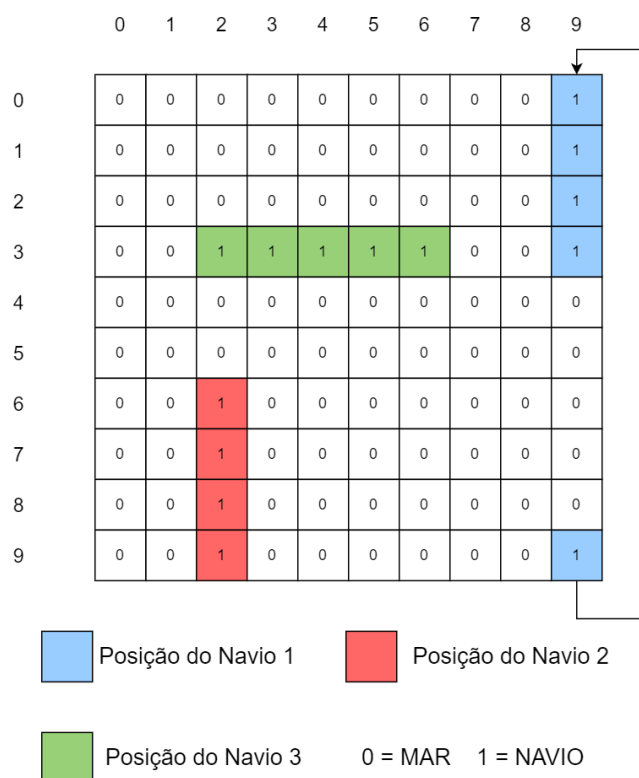


Figura 2: Exemplo de navio que dá a volta verticalmente na matriz.

bombardeiros para atacar, simultaneamente, todas as posições em uma linha ou coluna da matriz, como mostrado na Figura 3. Note que, se um ataque errado ocorrer antes do bombardeio ser chamado, o ataque especial fica invalidado, precisando de outros dois acertos consecutivos antes de ser chamado. Os oficiais dos exércitos da Terra da Computação não são muito inteligentes, e frequentemente pedem ataques especiais que não podem ser empregados ainda; neste caso, o turno do exército solicitante é desperdiçado e o ataque é dito mal sucedido. Por fim, vale lembrar que, apesar de inimigos, os Algoritmistas e os Estruturistas seguem à risca a etiqueta do campo de batalha e, por isso, o número de ataques de cada lado é sempre o mesmo.

### 3 Objetivo

Faça um programa em C que receba um histórico completo de um confronto marítimo entre os Algoritmistas e Estruturistas, referente a uma matriz com dimensões  $10 \times 10$  e identifique o resultado da batalha.

### 4 Descrição da Entrada

A entrada possui o seguinte formato: a primeira linha contém o único inteiro  $N$ , que representa o número de navios do exército Algoritmista. As próximas  $N$  linhas identificam o tamanho  $T$ , orientação (vertical ou horizontal)  $o$ , e as coordenadas (linha  $x$  e coluna  $y$ ) da ponta dos navios. Por exemplo, se lermos a linha `5H2`, então temos um navio de tamanho 5, posicionados horizontalmente na matriz, com sua ponta esquerda localizada na célula  $(3, 2)$  e sua ponta direita na célula  $(3, 6)$ . Outro exemplo: se lermos a linha `4V2`, então temos um navio de tamanho 4, disposto verticalmente, com a ponta superior na célula  $(6, 2)$  e a ponta inferior na célula  $(9, 2)$ . Em outras palavras, considere que todo navio vertical sempre “cresce” para baixo e que todo navio horizontal “cresce”, necessariamente, para a direita; reveja as Figuras 1 e 2 para esses e outros exemplos. Após a descrição dos navios do primeiro exército, temos uma nova linha com um inteiro  $M$ , representando o número de navios do exército Estruturista. Seguem então  $M$  linhas descrevendo os navios do segundo exército, seguindo a mesma semântica do primeiro exército.

Após essas  $N + M + 2$  linhas, temos uma nova linha contendo um inteiro  $A$ , que representa o número de ataques totais do confronto; ataques são realizados alternadamente, com os Algoritmistas sempre atacando primeiro: Algoritmistas  $\rightarrow$  Estruturistas  $\rightarrow$  Algoritmistas  $\rightarrow$  Estruturistas  $\rightarrow \dots$ ). O primeiro elemento de cada uma das  $A$  linhas seguintes é um caractere  $c$ , que representa o tipo de ataque. Se  $c == n$ , o exército atual está executando um ataque normal e, portanto, seguem dois inteiros  $x, y$ , que representam, respectivamente, a linha e coluna da célula atacada. Por outro lado, se  $c == e$ , então um ataque especial foi solicitado; nesse caso, segue um caractere,  $v$  ou  $h$ , indicando se o ataque é na vertical ou horizontal, respectivamente, seguido de um inteiro  $x$ , que corresponde à coluna ou linha a ser atacada. Lembre-se que o ataque pode não ser executado caso suas condições não sejam satisfeitas.

- $1 \leq N, M \leq 20$
- $1 \leq T \leq 5$

- $0 \leq A \leq 200$
- $0 \leq \text{numero de ataques especiais por exército} \leq 5$

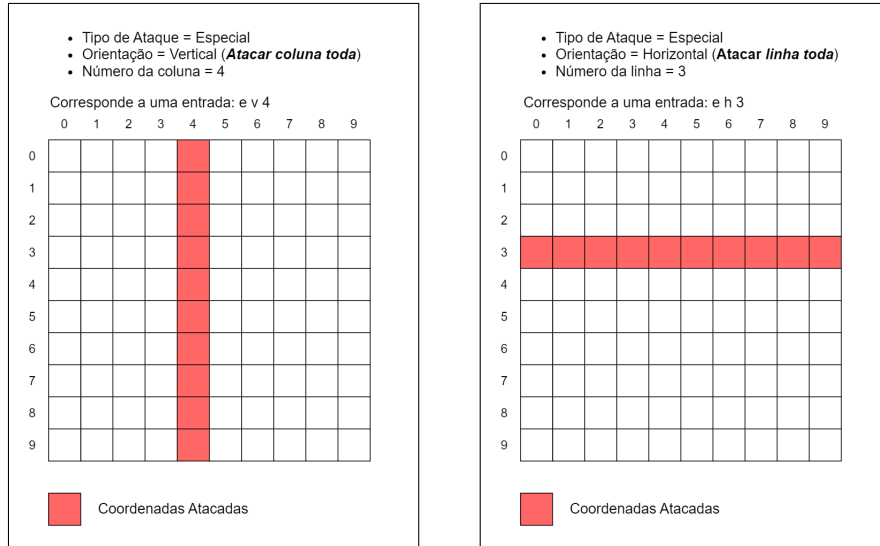


Figura 3: Ataques especiais.

Lembre-se que o ataque especial é opcional e não precisa ser usado necessariamente após 2 (ou mais) acertos consecutivos. A ideia é que o ataque especial tenha um contador, começando em 0 e que com cada acerto seja incrementado, passando a ser usável a partir de 2 acertos. Quando o ataque especial é usado, o contador deve ser reiniciado, independentemente de o ataque acertar ou não algum alvo. Quando um ataque especial é usado em um momento indevido (sem 2 acertos consecutivos), o exército que fez a jogada simplesmente perde a vez.

## 5 Descrição da Saída

A saída consiste de três partes: a matriz dos navios Algoritmistas seguida de uma linha em branco, a matriz dos Estruturistas, seguida de uma linha em branco, e um inteiro  $S$ , que deve ser igual a 0 caso o confronto tenha terminado em empate, 1 caso a vitória seja Algoritmista, ou 2 se a vitória for Estruturista.

Para a impressão das matrizes, siga as instruções abaixo:

## 5.1 Impressão das matrizes

### Impressão da matriz na tela com numeração e formatação

```
1  printf("_ _|");
2  for (int j = 0; j < 10; j++) {
3      printf("_%d", j);
4  }
5  printf("\n");
6  printf("--+");
7  for (int j = 0; j < 10-3; j++) {
8      printf("---");
9  }
10 putchar('\n');
11
12 for (int i = 0; i < 10; ++i) {
13     printf("%d_|", i);
14     for (int j = 0; j < 10; ++j) {
15         printf("_%d", player[i][j]);
16     }
17     putchar('\n');
18 }
19 putchar('\n');
```

## 5.2 Exemplo de impressão

```
1  | 0 1 2 3 4 5 6 7 8 9
2  --+-----
3  0 | 0 0 0 0 0 0 0 0 0 0
4  1 | 0 0 0 0 0 0 0 0 0 0
5  2 | 0 0 0 0 0 0 0 0 0 0
6  3 | 0 0 0 0 0 0 0 0 0 0
7  4 | 0 0 0 0 0 0 0 0 0 0
8  5 | 0 0 0 0 0 0 0 0 0 0
9  6 | 0 0 0 0 0 0 0 0 0 0
10 7 | 0 0 0 0 0 0 0 0 0 0
11 8 | 0 0 0 0 0 0 0 0 0 0
12 9 | 0 0 0 0 0 0 0 0 0 0
```

O tabuleiro irá iniciar com apenas 0 que indicará o *mar*. Todo barco será inserido na matriz com o valor 1. Quando o navio é atacado ele volta a ser considerado mar, ou seja, muda para o valor 0.

**Não escreva na saída padrão desnecessariamente.** Não é necessário escrever mensagens solicitando a entrada do usuário, por exemplo. Também não se deve imprimir estados intermediários das matrizes dos exércitos, **somente o estado final**. Se essa instrução não for seguida, seu trabalho pode ser zerado.

Para ajudar na visualização seguem alguns exemplos:

## 5.3 Exemplo 1

Número de navios Algoritmistas = 2

1º navio com Tamanho = 2, Orientação = v, Linha Coluna = 7 9

2º navio com Tamanho = 3, Orientação = h, Linha Coluna = 5 6

Número de navios Estruturistas = 1

1º navio com tamanho = 5, orientação = v, Linha Coluna = 2 4

Número de Ataques = 12

Algoritmistas: Tipo de Ataque = n, Linha Coluna = 1 1  
 Estruturistas: Tipo de Ataque = n, Linha Coluna = 3 1  
 Algoritmistas: Tipo de Ataque = n, Linha Coluna = 2 4  
 Estruturistas: Tipo de Ataque = n, Linha Coluna = 5 6  
 ...

Somente parte de inserir navios

```

1 2
2 2 v 7 9
3 3 h 5 6
4 1
5 5 v 2 4
  
```

Matriz Inicial de cada exército (NÃO DEVE SER IMPRESSA)

```

1  | 0 1 2 3 4 5 6 7 8 9
2  --+-----
3  0 | 0 0 0 0 0 0 0 0 0 0
4  1 | 0 0 0 0 0 0 0 0 0 0
5  2 | 0 0 0 0 0 0 0 0 0 0
6  3 | 0 0 0 0 0 0 0 0 0 0
7  4 | 0 0 0 0 0 0 0 0 0 0
8  5 | 0 0 0 0 0 0 1 1 1 0
9  6 | 0 0 0 0 0 0 0 0 0 0
10 7 | 0 0 0 0 0 0 0 0 0 1
11 8 | 0 0 0 0 0 0 0 0 0 1
12 9 | 0 0 0 0 0 0 0 0 0 0
13
14  | 0 1 2 3 4 5 6 7 8 9
15  --+-----
16 0 | 0 0 0 0 0 0 0 0 0 0
17 1 | 0 0 0 0 0 0 0 0 0 0
18 2 | 0 0 0 0 1 0 0 0 0 0
19 3 | 0 0 0 0 1 0 0 0 0 0
20 4 | 0 0 0 0 1 0 0 0 0 0
21 5 | 0 0 0 0 1 0 0 0 0 0
22 6 | 0 0 0 0 1 0 0 0 0 0
23 7 | 0 0 0 0 0 0 0 0 0 0
24 8 | 0 0 0 0 0 0 0 0 0 0
25 9 | 0 0 0 0 0 0 0 0 0 0
  
```

Somente parte dos Ataques

```

1 12          // Número de Ataques totais
2
3 n 1 1      // Errou
4 n 3 1      // Errou
5
6 n 2 4      // Acertou
7 n 5 6      // Acertou
8
9 n 2 5      // Errou
10 n 5 7     // Acertou
11
12 n 1 4     // Errou
13 n 5 8     // Acertou
14
15 n 3 4     // Acertou
16 n 7 3     // Errou
17
18 n 4 4     // Acertou
19 n 8 2     // Errou
  
```

Perceba que na 4.<sup>a</sup> rodada o exército dos Estruturistas poderia ter usado um ataque especial, visto que já tinham acertado duas vezes consecutivas seu adversário. Do mesmo modo, se acontecesse uma 7.<sup>a</sup> rodada, os Algoritmistas poderiam escolher usar um ataque especial também.

### Entrada — EX1

```

1 2
2 2 v 7 9
3 3 h 5 6
4 1
5 5 v 2 4
6 12
7 n 1 1
8 n 3 1
9 n 2 4
10 n 5 6
11 n 2 5
12 n 5 7
13 n 1 4
14 n 5 8
15 n 3 4
16 n 7 3
17 n 4 4
18 n 8 2

```

### Saída — EX1

```

1      | 0 1 2 3 4 5 6 7 8 9
2  --+-----
3  0 | 0 0 0 0 0 0 0 0 0 0
4  1 | 0 0 0 0 0 0 0 0 0 0
5  2 | 0 0 0 0 0 0 0 0 0 0
6  3 | 0 0 0 0 0 0 0 0 0 0
7  4 | 0 0 0 0 0 0 0 0 0 0
8  5 | 0 0 0 0 0 0 0 0 0 0
9  6 | 0 0 0 0 0 0 0 0 0 0
10 7 | 0 0 0 0 0 0 0 0 0 1
11 8 | 0 0 0 0 0 0 0 0 0 1
12 9 | 0 0 0 0 0 0 0 0 0 0
13
14      | 0 1 2 3 4 5 6 7 8 9
15  --+-----
16 0 | 0 0 0 0 0 0 0 0 0 0
17 1 | 0 0 0 0 0 0 0 0 0 0
18 2 | 0 0 0 0 0 0 0 0 0 0
19 3 | 0 0 0 0 0 0 0 0 0 0
20 4 | 0 0 0 0 0 0 0 0 0 0
21 5 | 0 0 0 0 1 0 0 0 0 0
22 6 | 0 0 0 0 1 0 0 0 0 0
23 7 | 0 0 0 0 0 0 0 0 0 0
24 8 | 0 0 0 0 0 0 0 0 0 0
25 9 | 0 0 0 0 0 0 0 0 0 0
26
27 0 // Empate já que sobrou navio de cada lado
28 // (comentários apenas para fins ilustrativos)

```



## 5.4 Exemplo 2

Somente parte de inserir navios

```
1 2
2 5 v 9 2
3 5 h 6 5
4 2
5 5 h 4 8
6 4 v 5 5
```

Matriz Inicial de cada exército

```
1 | 0 1 2 3 4 5 6 7 8 9
2 --+-----
3 0 | 0 0 1 0 0 0 0 0 0 0
4 1 | 0 0 1 0 0 0 0 0 0 0
5 2 | 0 0 1 0 0 0 0 0 0 0
6 3 | 0 0 1 0 0 0 0 0 0 0
7 4 | 0 0 0 0 0 0 0 0 0 0
8 5 | 0 0 0 0 0 0 0 0 0 0
9 6 | 0 0 0 0 0 1 1 1 1 1
10 7 | 0 0 0 0 0 0 0 0 0 0
11 8 | 0 0 0 0 0 0 0 0 0 0
12 9 | 0 0 1 0 0 0 0 0 0 0
13
14 | 0 1 2 3 4 5 6 7 8 9
15 --+-----
16 0 | 0 0 0 0 0 0 0 0 0 0
17 1 | 0 0 0 0 0 0 0 0 0 0
18 2 | 0 0 0 0 0 0 0 0 0 0
19 3 | 0 0 0 0 0 0 0 0 0 0
20 4 | 1 1 1 0 0 0 0 0 1 1
21 5 | 0 0 0 0 0 1 0 0 0 0
22 6 | 0 0 0 0 0 1 0 0 0 0
23 7 | 0 0 0 0 0 1 0 0 0 0
24 8 | 0 0 0 0 0 1 0 0 0 0
25 9 | 0 0 0 0 0 0 0 0 0 0
```

Somente parte dos Ataques

```
1 13 // Número de Ataques totais
2
3 n 4 0 // Acertou
4 n 3 1 // Errou
5
6 n 4 1 // Acertou
7 n 3 2 // Acertou
8
9 e h 4 // Acertou ataque especial
10 n 2 2 // Acertou
11
12 n 1 2 // Errou
13 e v 2 // Acertou ataque especial
14
15 n 8 5 // Acertou
16 n 6 7 // Acertou
17
18 n 7 5 // Acertou
19 n 5 7 // Errou
20
21 e v 5 // Acertou ataque especial
```

### Entrada — EX2

```

1 2
2 5 v 9 2
3 5 h 6 5
4 2
5 5 h 4 8
6 4 v 5 5
7 13
8 n 4 0
9 n 3 1
10 n 4 1
11 n 3 2
12 e h 4
13 n 2 2
14 n 1 2
15 e v 2
16 n 8 5
17 n 6 7
18 n 7 5
19 n 5 7
20 e v 5

```

### Saída — EX2

```

1      | 0 1 2 3 4 5 6 7 8 9
2  --+-----
3  0 | 0 0 0 0 0 0 0 0 0 0
4  1 | 0 0 0 0 0 0 0 0 0 0
5  2 | 0 0 0 0 0 0 0 0 0 0
6  3 | 0 0 0 0 0 0 0 0 0 0
7  4 | 0 0 0 0 0 0 0 0 0 0
8  5 | 0 0 0 0 0 0 0 0 0 0
9  6 | 0 0 0 0 0 1 1 0 1 1
10 7 | 0 0 0 0 0 0 0 0 0 0
11 8 | 0 0 0 0 0 0 0 0 0 0
12 9 | 0 0 0 0 0 0 0 0 0 0
13
14     | 0 1 2 3 4 5 6 7 8 9
15  --+-----
16 0 | 0 0 0 0 0 0 0 0 0 0
17 1 | 0 0 0 0 0 0 0 0 0 0
18 2 | 0 0 0 0 0 0 0 0 0 0
19 3 | 0 0 0 0 0 0 0 0 0 0
20 4 | 0 0 0 0 0 0 0 0 0 0
21 5 | 0 0 0 0 0 0 0 0 0 0
22 6 | 0 0 0 0 0 0 0 0 0 0
23 7 | 0 0 0 0 0 0 0 0 0 0
24 8 | 0 0 0 0 0 0 0 0 0 0
25 9 | 0 0 0 0 0 0 0 0 0 0
26
27 1 // Algoritmistas venceram (comentário ilustrativo)

```

## 7 Exemplos de Entradas

### EX1 — Entrada

```

1 4
2 2 v 2 2
3 1 h 9 0
4 2 h 0 9
5 1 v 6 1
6 3
7 2 h 6 1
8 2 v 3 3
9 1 v 1 0
10 9
11 n 1 0
12 n 0 9
13 n 4 3
14 n 1 9
15 n 6 2
16 n 0 0
17 n 3 3
18 n 2 1
19 n 6 1

```

### EX1 — Saída

```

  | 0 1 2 3 4 5 6 7 8 9
--+-+-----
0 | 0 0 0 0 0 0 0 0 0 0
1 | 0 0 0 0 0 0 0 0 0 0
2 | 0 0 1 0 0 0 0 0 0 0
3 | 0 0 1 0 0 0 0 0 0 0
4 | 0 0 0 0 0 0 0 0 0 0
5 | 0 0 0 0 0 0 0 0 0 0
6 | 0 1 0 0 0 0 0 0 0 0
7 | 0 0 0 0 0 0 0 0 0 0
8 | 0 0 0 0 0 0 0 0 0 0
9 | 1 0 0 0 0 0 0 0 0 0

  | 0 1 2 3 4 5 6 7 8 9
--+-+-----
0 | 0 0 0 0 0 0 0 0 0 0
1 | 0 0 0 0 0 0 0 0 0 0
2 | 0 0 0 0 0 0 0 0 0 0
3 | 0 0 0 0 0 0 0 0 0 0
4 | 0 0 0 0 0 0 0 0 0 0
5 | 0 0 0 0 0 0 0 0 0 0
6 | 0 0 0 0 0 0 0 0 0 0
7 | 0 0 0 0 0 0 0 0 0 0
8 | 0 0 0 0 0 0 0 0 0 0
9 | 0 0 0 0 0 0 0 0 0 0

1          // Resultado da Batalha = Algoritmistast venceram

```

## EX2 — Entrada

```

1 4
2 3 h 6 9
3 5 v 2 5
4 4 v 1 1
5 1 h 8 2
6 4
7 4 v 5 0
8 5 v 3 6
9 3 h 3 2
10 5 v 1 9
11 20
12 n 1 3
13 n 6 0
14 n 5 3
15 n 6 1
16 n 3 2
17 e h 6
18 n 3 3
19 n 4 1
20 n 3 4
21 n 3 1
22 n 3 5
23 e h 1
24 n 8 2
25 n 2 5
26 n 5 0
27 n 8 2
28 n 4 0
29 e v 5
30 n 3 0
31 n 2 1

```

## EX2 — Saída

```

  | 0 1 2 3 4 5 6 7 8 9
--+-+-----
0 | 0 0 0 0 0 0 0 0 0 0
1 | 0 0 0 0 0 0 0 0 0 0
2 | 0 0 0 0 0 0 0 0 0 0
3 | 0 0 0 0 0 0 0 0 0 0
4 | 0 0 0 0 0 0 0 0 0 0
5 | 0 0 0 0 0 0 0 0 0 0
6 | 0 0 0 0 0 0 0 0 0 0
7 | 0 0 0 0 0 0 0 0 0 0
8 | 0 0 0 0 0 0 0 0 0 0
9 | 0 0 0 0 0 0 0 0 0 0

  | 0 1 2 3 4 5 6 7 8 9
--+-+-----
0 | 0 0 0 0 0 0 0 0 0 0
1 | 0 0 0 0 0 0 0 0 0 1
2 | 0 0 0 0 0 0 0 0 0 1
3 | 0 0 0 0 0 0 1 0 0 1
4 | 0 0 0 0 0 0 1 0 0 1
5 | 0 0 0 0 0 0 1 0 0 1
6 | 1 0 0 0 0 0 1 0 0 0
7 | 1 0 0 0 0 0 1 0 0 0
8 | 1 0 0 0 0 0 0 0 0 0
9 | 0 0 0 0 0 0 0 0 0 0

```

```

2 // Resultado da Batalha = Vitoria dos Estruturistas.

```

### EX3 — Entrada

```

1 4
2 5 v 4 3
3 5 h 2 5
4 4 h 5 6
5 5 v 8 1
6 5
7 5 v 6 6
8 4 h 3 8
9 5 v 5 2
10 5 h 0 0
11 5 v 3 4
12 14
13 n 3 4
14 n 5 3
15 n 4 4
16 n 5 6
17 e v 4
18 e h 5
19 n 5 5
20 n 8 1
21 n 3 8
22 n 9 1
23 n 3 9
24 e v 1
25 e h 3
26 n 4 3

```

### EX3 — Saída

```

  | 0 1 2 3 4 5 6 7 8 9
--+-+-----
0 | 0 0 0 0 0 0 0 0 0 0
1 | 0 0 0 0 0 0 0 0 0 0
2 | 0 0 0 0 0 1 1 1 1 1
3 | 0 0 0 0 0 0 0 0 0 0
4 | 0 0 0 0 0 0 0 0 0 0
5 | 0 0 0 0 0 0 0 0 0 0
6 | 0 0 0 1 0 0 0 0 0 0
7 | 0 0 0 1 0 0 0 0 0 0
8 | 0 0 0 1 0 0 0 0 0 0
9 | 0 0 0 0 0 0 0 0 0 0

  | 0 1 2 3 4 5 6 7 8 9
--+-+-----
0 | 1 1 1 1 0 0 1 0 0 0
1 | 0 0 0 0 0 0 0 0 0 0
2 | 0 0 0 0 0 0 0 0 0 0
3 | 0 0 0 0 0 0 0 0 0 0
4 | 0 0 0 0 0 0 0 0 0 0
5 | 0 0 1 0 0 0 0 0 0 0
6 | 0 0 1 0 0 0 1 0 0 0
7 | 0 0 1 0 0 0 1 0 0 0
8 | 0 0 1 0 0 0 1 0 0 0
9 | 0 0 1 0 0 0 1 0 0 0

0 // Resultado da Batalha = Empate

```