MAC-115 Introdução à Computação para Ciências Exatas e Tecnologia
IO – Quarto Exercício-Programa – Entregar até 07/12/2007

Uma Batalha Oceanográfica

Este exercício-programa consiste na implementação de uma versão muuuito simplificada de um jogo inspirado na famosa Batalha Naval. Neste jogo, a missão do usuário (ou do jogador) do seu programa será atravessar, utilizando um barco a remo, uma determinada região retangular de um oceano, onde está ocorrendo uma batalha.

Outras embarcações estarão presentes nessa região, e podem ser dos seguintes tipos:

$$\textit{Submarino} \left(\text{S} \right), \quad \textit{Destroyer} \left(\text{DD} \right), \quad \textit{Cruzador} \left(\begin{array}{c} \text{CC} \\ \text{CC} \end{array} \right) \quad \text{e} \quad \textit{Porta-avião} \left(\text{PPPP} \right).$$

A região da batalha é descrita através de um mapa, que é representado no computador por uma matriz de caracteres, contendo informações sobre o conteúdo de cada posição da região, inclusive onde está localizada cada embarcação. Essa matriz pode ter no máximo 18 linhas e no máximo 20 colunas.

Veja um exemplo de um mapa que descreve uma região (com 15 linhas e 18 colunas):

Exemplo de um mapa da região da batalha

```
   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
1 | - | D | - | - | - | - | - | C | C | - | - | S | - | - | - | - | P | - |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
2 | - | D | - | - | - | S | - | C | C | - | - | - | - | D | - | - | P | - |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
3 | - | - | - | - | - | - | - | - | - ! - | - | - | - | - | D | - | - | P | - |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
4 | - | C | C | - | - | - | - | - | P | P | P | P | - | - | - | - | - | P | - |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
5 | - | C | C | - | - | - | - | - | - | - | - | - | - | - | - | S | - | - | - |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
6 | - | - | - | - | - | D | D | - | - | - | S | - | - | - | - | - | - | - | D |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
7 | S | - | - | - | - | - | - | - | - | - | - | - | - | C | C | - | - | - | - | D |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
8 | - | - | - | S | - | - | - | - | - | - | - | - | - | C | C | - | - | - | - | - |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
9 | - | - | - | - | - | - | - | - | S | - | - | - | - | - | - | - | - | C | C | - |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
10 | - | P | P | P | P | - | - | - | - | - | - | - | - | - | - | - | - | C | C | - |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
11 | - | - | - | - | - | - | - | - | - | C | C | - | - | - | - | - | - | - | - | - |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
12 | - | - | - | - | - | - | D | D | - | C | C | - | - | - | P | - | - | - | D | D |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
13 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | P | - | - | - | - | - |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
14 | D | D | - | - | - | - | - | - | - | - | - | - | - | - | P | - | - | - | - | - |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
15 | - | - | - | - | - | - | - | S | - | - | - | D | D | - | - | P | - | - | - | S |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

No início da batalha, o barco deve ser posicionado em alguma coluna da primeira linha. Na matriz, a posição do barco deve ter 'B'.

O objetivo do jogador é conduzir o barco a remo da posição inicial na linha 1 até uma posição qualquer na última linha, através de remadas sucessivas. Mas, durante a travessia, o mar não estará tranqüilo. O jogador terá que desviar de obstáculos (embarcações) que existem no caminho e, além disso, estará sujeito, a cada remada, a três tiros que podem acertar o seu barco. (Se isto acontecer, o jogador se deu mal e o jogo terá acabado!)

Os tiros são disparados aleatoriamente pelo computador e são tão fortes que se um tiro acertar parte de uma embarcação, esta embarcação deve afundar totalmente, e o local fica livre para a passagem do seu barco.

Assim, cada jogada consiste em:
(a) o jogador tenta movimentar o barco;
(b) três tiros são disparados sobre a região.

O jogo termina quando ocorrer uma das duas situações a seguir:
(a) o jogador conduziu o barco até alguma posição da última linha, e está a salvo;
(b) o barco foi atingido por um tiro.
Obs.: O barco pode ser atingido por um tiro estando na última linha.

Escreva um programa em C (padrão ANSI) para simular este jogo. O seu programa deve definir e utilizar, obrigatoriamente, pelo menos as oito funções descritas a seguir. Para aquelas que não têm o protótipo, decida quais parâmetros são necessários, e **não** utilize variáveis globais.

Alguns dos protótipos abaixo fazem uso de constantes definidas a seguir:

```
#define MAXLIN   20
#define MAXCOL   22
#define MAXNOME  31
```

1. `void leia_ate_fim_linha (FILE *arq);`

   Lê caracteres do arquivo associado a *arq* até que um fim-de-linha ('\n') seja lido.

2. `void leia_mapa (int *plin, int *pcol, char mapa[][MAXCOL]);`

   Lê de um dado arquivo dois inteiros *nlin* e *ncol*, e uma matriz de caracteres com *nlin* linhas e *ncol* colunas (representando um mapa de uma região como no modelo descrito anteriormente), e armazena em $*plin$ o valor de *nlin*, em $*pcol$ o valor de *ncol*, e em *mapa* a matriz de caracteres lida.

   A primeira linha desse arquivo deve conter os inteiros *nlin* e *ncol*, e cada uma das *nlin* linhas seguintes desse arquivo deve conter uma linha da matriz.

   O nome desse arquivo deve ser fornecido pelo usuário, e tudo que se refere ao arquivo de entrada deve ficar restrito a esta função. Veja a seguir.

```
void leia_mapa (int *plin, int *pcol, char mapa[][MAXCOL])
{
  FILE *arqentrada;
  char nomearq[MAXNOME];
  int nlin, ncol, i, j;

  printf ("Digite o nome (maximo %d caracteres) do arquivo de entrada :\n", MAXNOME-1);
  scanf ("%s", nomearq);
  arqentrada = fopen (nomearq, "r");
  if (arqentrada == NULL)  {
    printf ("ERRO: nao foi possivel abrir arquivo %s\n", nomearq);
    exit (-1);
  }
```

```
  /* ... */
  fscanf (arqentrada, "%d %d", &nlin, &ncol);
  leia_ate_fim_linha (arqentrada);
  /* ... */
  for (i = 1; i <= nlin; i++)   {
     for (j = 1; j <= ncol; j++)
        fscanf (arqentrada, "%c", &mapa[i][j]);
     leia_ate_fim_linha (arqentrada);
  }
  fclose (arqentrada);
  /* ... */
}

void leia_ate_fim_linha (FILE *arq)
{
  char carac;
  fscanf (arq, "%c", &carac);
  while (carac != '\n')
     fscanf (arq, "%c", &carac);
}
```

3. `void escreva_mapa_tela (char mapa[][MAXCOL], int nlin, int ncol);`

   Recebe uma matriz *mapa* com *nlin* linhas e *ncol* colunas, e escreve na tela o *mapa* (visível para o jogador), destacando cada uma de suas posições, e escreve também os índices de cada linha e coluna.

   Como o jogador não pode saber onde estão as embarcações, inicialmente, todas as posições devem exibir o caractere '.'. Nas situações seguintes, o jogador pode ver a posição ocupada pelo barco (que contém o caractere 'B', ou o caractere '+' se o barco foi atingido por um tiro nesta jogada), e também as posições que eram ocupadas por alguma embarcação que foi atingida por um tiro (isto é, que contêm o caractere '$*$'). As demais posições continuam exibindo o caractere '.'.

4. `void escreva_mapa_arquivo (FILE *arq, char mapa[][MAXCOL], int nlin, int ncol);`

   Recebe uma matriz *mapa* com *nlin* linhas e *ncol* colunas, e escreve *mapa* no arquivo associado a *arq*. Escreve também os índices de cada linha e coluna da matriz.

5. `int coluna_inicial_barco (char mapa[][MAXCOL], int ncol);`

   Lê (via teclado) um índice de coluna para a posição inicial do barco, marca com 'B' esta posição na matriz *mapa*, e devolve este índice. O inteiro *ncol* representa o número de colunas da matriz *mapa*.

   Como o barco não pode começar numa posição ocupada por alguma embarcação, a leitura deve ser feita até encontrar uma posição que não esteja ocupada.

6. `int rema_barco (???);`

   Vai tentar movimentar o barco ou para baixo (na direção da última linha) ou na horizontal, tomando cuidado para não ir para uma posição ocupada por alguma embarcação ou uma posição inválida da matriz.

   O tipo do movimento deve ser lido (apenas uma vez) do teclado; ou seja, o jogador deve digitar um dos seguintes caracteres: 'b', 'd' ou 'e', indicando, respectivamente, que deseja movimentar o barco para baixo, para a direita ou para a esquerda.

   Se for possível movimentar o barco, a função deve atualizar na matriz a posição do barco (com 'B'), atualizar o índice de linha ou de coluna da posição do barco, e deve devolver 1; em caso contrário, deve devolver 0.

7. `int num_aleatorio (int k);`

   Esta função recebe um inteiro k, e devolve um inteiro no intervalo $[1, k]$ fornecido pela seguinte expressão `(int)(1 + (rand()/(RAND_MAX+1.0))*k)` .

8. `??? dispara_tiros (???);`

   Esta função deve determinar as posições dos três tiros a serem disparados pelo computador, e escrever (tela e arquivo) a posição de cada tiro e as mensagens correspondentes ao efeito de cada tiro.

   Se um tiro acertar uma embarcação, deve-se especificar o seu tipo, e atualizar a matriz representando o mapa, atribuindo-se o caractere '$*$' a todas as posições ocupadas pela embarcação que foi afundada. Se um tiro acertar o barco, deve atualizar na matriz a posição do barco com o caractere '+'. Se a posição da matriz atingida por um tiro tiver o caractere '$-$' ou '$*$', ela não deverá ser alterada.

   Após cada tiro, deve escrever (tela e arquivo) a matriz atualizada.

   Para determinar a posição (isto é, os índices de linha e de coluna) de um tiro, chame duas vezes a função `num_aleatorio`, uma para cada índice.

Observações importantes:

(1) Suponha que as embarcações podem estar posicionadas ou na horizontal ou na vertical, e que elas nunca se encostam, mas o barco pode encostar em qualquer embarcação.

Além disso, considere que as embarcações não se movimentam durante o jogo.

(2) Quando terminar o jogo, imprima uma mensagem descrevendo o que aconteceu com o barco.

(3) Este program deve gerar duas saídas.

- Saída para a tela, escrevendo o mapa que o usuário pode ver. Este mapa deve ser mostrado após cada movimento do barco, e também após cada tiro, com mensagens correspondentes.
- Saída para um arquivo, onde deverão ser escritos a matriz cada vez que ela é alterada, e as mensagens referentes aos tiros e aos movimentos do barco.

O arquivo de saída deve conter um cabeçalho com pelo menos o seu nome e número USP.

(4) A função `rand` devolve o próximo inteiro de uma seqüência de números inteiros pseudo-aleatórios no intervalo de 0 a `RAND_MAX`.

A constante `RAND_MAX` representa o maior inteiro correspondente ao tipo *int*.

Para que seqüências diferentes sejam geradas cada vez que o programa é executado, pode-se utilizar a função `srand`, que estabelece o argumento como sendo a semente para a seqüência de números pseudo-aleatórios que serão devolvidos pelas chamadas subseqüentes da função `rand`.

A semente pode ser gerada, por exemplo, utilizando a função `time`. Para inicializar a semente, chame a função `srand`, uma única vez, no início da função principal, da seguinte forma:

```
srand ((unsigned int) time(NULL));
```

Para utilizar em seu programa as funções `rand` e `srand`, e a constante `RAND_MAX`, inclua o arquivo *stdlib.h*, e para utilizar a função `time`, inclua o arquivo *time.h*.

(5) Não altere os nomes das variáveis, os nomes das funções, e os nomes e a ordem dos parâmetros das funções, que aparecem neste enunciado.

(6) Defina e utilize outras constantes. Por exemplo, para cada uma das embarcações, para cada movimento do barco, para a água, etc.

(7) Leia também as observações (exceto a 5) do ep1.