

### Trabalho Prático 1 - Simulador RAMSES

Escrever um programa que prepara um campo de 64 posições dispostas em 8 linhas e 8 colunas para jogar “Campo Minado” (para quem não conhece o jogo, consultar o seguinte URL: [http://pt.wikipedia.org/wiki/Campo\\_minado](http://pt.wikipedia.org/wiki/Campo_minado)), registrando em cada posição do campo qual a quantidade de minas que estão escondidas nas posições em volta da mesma.

O programa recebe o campo apenas com a indicação das posições em que existem minas escondidas (representadas pelo valor -1) e as demais posições contendo o valor zero. Sempre existem exatamente 8 minas escondidas no campo, mas por questões de segurança, as minas estarão escondidas apenas nas 16 posições mais internas do campo (para evitar que alguém que esteja passeando em volta do campo caia acidentalmente sobre uma mina escondida, por exemplo ;-)). Em outras palavras, as duas posições mais externas em cada linha ou coluna sempre estarão livres de minas.

#### Definições

O campo está mapeado para as posições 192 a 255 da memória, linha por linha. A primeira linha do campo é mapeada para as posições 192 a 199, a segunda linha para as posições 200 a 207 e assim sucessivamente, como mostrado na figura à esquerda, a seguir. Nesta figura, a área salientada em cinza corresponde às 16 posições onde pode haver minas escondidas. Nas demais 48 posições nunca haverá minas.

192	193	194	195	196	197	198	199
200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215
216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231
232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247
248	249	250	251	252	253	254	255

Mapeamento do campo na memória

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	-1	0	-1	0	0
0	0	0	-1	0	-1	0	0
0	0	0	-1	-1	-1	0	0
0	0	0	0	-1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Exemplo de conteúdo inicial do campo

No início da execução do programa, cada posição poderá conter o valor -1, que representa uma mina escondida ou o valor 0, como mostrado na figura à direita, acima. Nesta figura, as posições salientadas em cinza são as que contêm minas escondidas.

O programa deverá então fazer a contagem de minas escondidas em volta de cada posição do campo, e armazenar, em cada posição, um valor de 0 a 8 que indica quantas minas estão escondidas em torno dela. É importante lembrar que algumas das posições onde não podem ser escondidas minas podem terminar com um valor diferente de zero quando existem minas ao lado delas (ver a segunda linha e a penúltima coluna, no exemplo a seguir).

A figura abaixo mostra quais deveriam ser os valores finais no campo, quando as 8 minas estiverem escondidas nas posições salientadas em cinza.

0	0	0	0	0	0	0	0
0	0	1	1	2	1	1	0
0	0	2	-1	4	-1	2	0
0	0	3	-1	7	-1	3	0
0	0	2	-1	-1	-1	2	0
0	0	1	3	-1	3	1	0
0	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0

Contagem de minas escondidas no campo

### A implementação

O programa receberá uma matriz 8x8 representando as 64 posições do campo minado. Esta matriz ocupará as posições 192 a 255 da memória (1 byte para cada posição do campo minado). O programa deverá fazer a contagem da quantidade de minas que existe em volta de cada posição do campo (esta etapa é o primeiro passo para preparar o tabuleiro para o jogo “Campo Minado”). Cada posição do campo é representada por um byte. No final da execução, se o valor em uma posição for zero ou positivo, aquela posição está vazia (não existe uma mina escondida nela) e o valor nela contido (de 0 a 8) indica quantas minas estão escondidas em volta dela. As oito posições do campo em que existem minas escondidas continuarão contendo o valor -1.

O trabalho deve ser desenvolvido usando o montador Daedalus, pois a complexidade do algoritmo poderá tornar a depuração de sua solução muito demorada se usado diretamente o simulador. Os trabalhos serão corrigidos de forma automática, com 20 conjuntos de dados de teste diferentes. Portanto, devem ser observadas rigorosamente as seguintes especificações:

- o código do programa deve iniciar no endereço 0 da memória.
- a primeira instrução executável deve estar no endereço 0.
- os endereços de 191 a 255 devem ser reservados para a representação do campo minado, conforme definido acima.
- como o programa será executado 20 vezes de forma consecutiva (sem ser carregado de disco a cada vez), *é necessário inicializar as variáveis utilizadas no seu programa.*
- uma nova matriz será carregada para cada caso de teste.
- garanta-se que os dados de entrada estão corretos, segundo a definição acima.

O programa deverá ser entregue via Moodle, compactado (zip ou rar), incluindo uma cópia do arquivo **.mem** e uma do arquivo **.rad** – fonte para o Ramses no formato para o montador Daedalus, devidamente comentado e contendo a identificação do autor. Para os nomes dos arquivos, utilizar os mesmos nomes padronizados dos trabalhos de INF01107, ou seja, inicial do primeiro nome seguido do seu número de identificação.

**SUGESTÃO:** usar uma subrotina para fazer a contagem de minas em torno de uma determinada posição. A subrotina receberá como parâmetro o endereço da posição a ser analisada e devolverá a quantidade de minas encontradas em torno daquela posição. Por exemplo, se o endereço recebido for igual a 220, a subrotina vai analisar as 8 posições do campo em volta da posição 220 para determinar quantas minas estão escondidas nelas. No exemplo da página anterior, isto significa analisar a parte do campo mostrada na figura abaixo, na qual a posição salientada em cinza é o byte 220 da memória. Neste caso, o resultado no byte 220 deverá ser 7.

-1	0	-1
-1	?	-1
-1	-1	-1

#### **DICA (uso do montador Daedalus)**

Ao codificar o programa, lembrar que o montador Daedalus permite a definição do valor de vários bytes em uma mesma linha. O código abaixo é a definição do campo minado para o exemplo da página 1:

```
ORG 192
Campo: DAB 0, 0, 0, 0, 0, 0, 0, 0, 0
        DAB 0, 0, 0, 0, 0, 0, 0, 0, 0
        DAB 0, 0, 0,-1, 0,-1, 0, 0, 0
        DAB 0, 0, 0,-1, 0,-1, 0, 0, 0
        DAB 0, 0, 0,-1,-1,-1, 0, 0, 0
        DAB 0, 0, 0, 0,-1, 0, 0, 0, 0
        DAB 0, 0, 0, 0, 0, 0, 0, 0, 0
        DAB 0, 0, 0, 0, 0, 0, 0, 0, 0
```

<b>Data final de Entrega: 23/04/2010, 18 horas, via Moodle</b>
--