



Instruções:

Trabalho em dupla

Data 1 entrega: Menu de seleção de tamanho e função que mostra o estado do jogo (situação do tabuleiro) - 03/06/2022

Data 2a entrega: Função `insere_bombas` e menu das jogadas (incluindo a leitura de linha e coluna): 24/07/2022

Data de entrega final: 17/08/2022

Entregar o arquivo `.asm` ou `.s` exclusivamente através do ambiente Moodle.

O nome do arquivo deve ter o nome dos integrantes da dupla

Descrição:

Implemente um programa que permita ao usuário jogar Campo Minado.

O programa deve ser implementado utilizando o conjunto de instruções RV32I e RV32M do RISC-V a ser executado no simulador RARS.

O programa permite ao usuário jogar campo minado em um 'tabuleiro' de tamanho 8x8, 10x10 ou 12x12. Internamente o tabuleiro é uma matriz (chamada `campo`) com tamanho 12x12 e é composta de números inteiros a qual deve ser inicializada com zeros pelo programador. A matriz `campo` representa as posições do campo minado, onde cada posição (linha x coluna) pode ou não conter uma bomba.

O programa **deve** ser implementado com uma função principal (`main`) e pelo menos duas outras funções:

- função `calcula_bombas` com o seguinte protótipo:

```
void calcula_bombas(int * campo[], int num_linhas);
```

- função `mostra_campo` (cada aluno pode definir o protótipo da mesma);

Os parâmetros de `calcula_bombas` devem ser passado nos registradores `a0` e `a1`, respectivamente.

Após inicializar as variáveis de controle o programa deve chamar uma função chamada `insere_bombas`, com o seguinte protótipo:

```
void insere_bombas(int * campo[], int num_linhas);
```

A função `insere_bombas` não precisa ser codificada pelo aluno pois será fornecida pelo professor. Os parâmetros de `insere_bombas` deve ser passado nos registradores `a0` e `a1`, respectivamente. Esta função colocará as bombas na matriz `campo`.

As bombas são representadas pelo valor 9 armazenado na posição (linha x coluna). Assim, na matriz `campo` as casas que possuírem o valor 9 possuem bomba naquela posição. O programa deve permitir inserir bandeiras nas casas (linha x coluna) impedindo que a casa seja aberta posteriormente (no caso, abrir o campo naquela posição).

Inicialmente o programa deve apresentar uma interface que permite ao usuário escolher o tamanho do tabuleiro: (1) 8x8; (2) 10x10; (3) 12x12. Após a seleção de tamanho deve ser chamada a função `insere_bombas` e então deve-se iniciar o jogo, apresentando para o usuário uma interface que, a cada jogada permite: (a) abrir uma posição (mostrar o número de bombas presentes no entorno) e (b) inserir uma bandeira ou retirar uma bandeira em uma posição. Após escolher entre as opções (a) ou (b) o usuário deve digitar o número da linha e da coluna que pretende jogar (bandeira ou abrir a posição, conforme escolha anterior). Após a jogada deve-se mostrar o estado atual do jogo (imprimir o tabuleiro do jogo com as casas abertas e as bandeiras), utilizando a função `mostra_campo`.

Caso tenha uma bomba na posição jogada o programa deve indicar que uma bomba explodiu e o usuário perdeu (mostrando na interface a própria matriz campo, com a quantidade de bombas no entorno das posições, as posições das bombas e as posições das bandeiras); Caso não tenha uma bomba naquela posição o programa deve mostrar a situação do campo (mostrando a quantidade de bombas no entorno das posições já jogadas e as bandeiras já colocadas); Caso a posição escolhida para abrir tenha uma bandeira uma mensagem informado que a posição não pode ser aberta (jogada inválida).

Avaliação:

A avaliação do trabalho levará em consideração os seguintes aspectos:

- funcionalidades implementadas (60%): menus; abrir posição; inserir bandeira; retirar bandeira; mostrar o campo com posições abertas e bandeiras; mostrar estado do jogo quando usuário ganha ou perde;
- codificação (15%): uso de chamadas de função; lógica de implementação; documentação do código;
- apresentação do trabalho (25%): clareza da explicação; entendimento do código fonte;
- o atraso de cada uma das entregas parciais terão desconto de 10% na nota final, por atraso.

Dicas e exemplo:

- Logo após a chamada da função `insere_bombas` chamar a função `calcula_bombas` para realizar a contagem de bombas vizinhas em cada posição e armazenar na matriz `campo` esta informação (veja figura abaixo);
- Utilizar a matriz `campo` para o controle e uma outra matriz para a interface com o usuário;
- Sinalizar as bandeiras inseridas somando 10 na quantidade de bombas na matriz `campo`;
- A cada jogada do usuário fazer as marcações apropriadas na matriz `campo` e na matriz da interface, por exemplo, copiando a quantidade de bombas da matriz `campo` para a matriz interface, na posição da jogada; somando 10 na posição onde a bandeira foi colocada e colocando a letra F na matriz da interface, etc;
- Exemplo de matriz `campo` e respectiva matriz de interface após algumas jogadas realizadas:

campo

0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
0	0	1	9	2	1	1	0
0	0	2	2	3	9	1	0
0	1	2	9	2	1	1	0
0	1	9	2	1	0	0	0
0	1	1	1	0	10	0	0
0	0	0	0	0	0	0	0



interface

-	-	-	-	-	-	-	-
-	-	1	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	3	-	-	-
-	-	-	-	-	-	-	-
-	-	-	1	-	-	-	0
-	-	-	-	-	F	-	-
-	-	-	-	-	-	-	-

Interface (text mode)

	0	1	2	3	4	5	6	7
0	-	-	-	-	-	-	-	-
1	-	-	1	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	3	-	-	-
4	-	-	-	-	-	-	-	-
5	-	-	-	1	-	-	-	0
6	-	-	-	-	-	F	-	-
7	-	-	-	-	-	-	-	-