

21010 - Arquitetura de Computadores

Enunciado

Tal como aconteceu no e-fólio A, vamos considerar o jogo do “Master Mind”. Repetem-se aqui as regras:

Considere o jogo de dois jogadores “Master Mind”. O Objetivo do jogo é um dos jogadores descobrir uma chave, definida pelo outro jogador. A chave é composta por quatro posições, numeradas de 0 a 3, e em cada posição é colocado um pino. Existem pinos de quatro cores: amarelo, vermelho, azul e verde. A chave pode conter cores repetidas. A chave é definida no início do jogo e mantém inalterada até ao final.

Na Figura 1 apresenta-se um exemplo de chave de jogo:

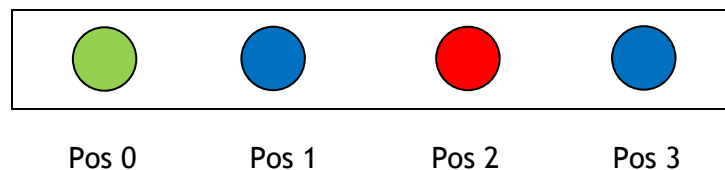


Figura 1 - Exemplo de uma chave do jogo do Master Mind.

O jogador que está a descobrir a chave dispõe de seis tentativas para tal. Em cada tentativa o jogador que tenta adivinhar a chave coloca 4 pinos de cores nas posições de 0 a 3. O jogador que definiu a chave dá então o resultado da tentativa, indicando o número de:

- Pinos pretos, ou seja número de pinos da cor certa na posição certa;
- Pinos brancos, ou seja o número de pinos da cor certa, mas colocados na posição errada.

Na Figura 2 apresenta-se uma tentativa e o respectivo resultado, considerando a chave da Figura 1:

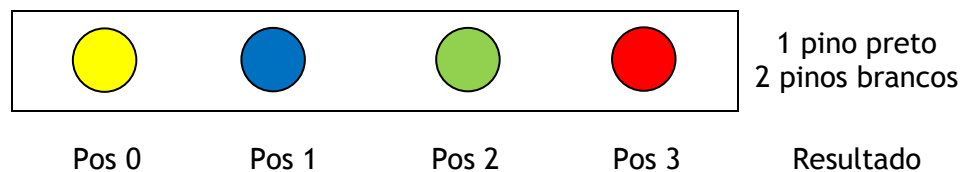


Figura 2 - Exemplo de uma tentativa e resultado, para a chave da Figura 1.

No resultado não há qualquer indicação das posições a que se referem os pinos.

Se o jogador consegue adivinhar a chave nas seis tentativas de que dispõe, ganha o jogo, e perde em caso contrário.

a) [1] Desenvolva no Assembly do P3 uma subrotina denominada **NPPretos** que coloque no registo R1 o número de pinos pretos de uma determinada tentativa de adivinhar a chave do jogo “Master Mind”. Considere que a chave se inicia na posição de memória 4000h, e que a tentativa se inicia na posição de memória 4008h. Para testar a subrotina desenvolvida, utilize a opção do simulador **Definições->Zona de memória** para ver no início da página o endereço 4000h. Para colocar cada cor quer na chave quer na tentativa utilize a opção do simulador **Depuração->Escreve Memória** indicando a posição de memória e a cor correspondente.

NOTA: Considere cada cor como um valor inteiro.

b) [1] Desenvolva no Assembly do P3 uma subrotina denominada **NPBranco**s que coloque no registo R1 o número de pinos brancos de uma dada tentativa de adivinhar a chave do jogo “Master Mind”.

c) [1] Desenvolva no Assembly do P3 uma subrotina denominada **Tentativa** que receba como parâmetros de entrada o número da tentativa seguido da posição de memória onde se encontra armazenada a primeira cor da chave (POS), e que escreva em:

POS + 4 (número de cores da chave)	Número de pinos pretos da tentativa
POS + 5	Número de pinos brancos da tentativa
POS + 7	Número da tentativa

Utilize nesta subrotina as duas subrotinas desenvolvidas nas alíneas a) e b).

d) [1] Desenvolva um programa no Assembly do P3 que simule um jogo de “Master Mind” com as seis tentativas, e que no final do jogo escreva na posição de memória 4038h o valor 0 se o jogador perdeu o jogo, ou o valor 1 se o jogador venceu o jogo. Caso o jogador tenha vencido o jogo na posição de memória seguinte deve ser escrita a tentativa em que ganhou. Essa posição deve ser limpa (valor 0) no caso do jogador perder o jogo.

Mantêm-se as posições de memória indicadas, no que diz respeito à chave e à primeira tentativa, devendo a segunda tentativa usar o mesmo formato de dados mas começando na posição 4010h, e assim sucessivamente até à 6ª tentativa.

O ficheiro **DadosMem.txt** pode ser utilizado para testar o programa. Este carrega uma chave e um conjunto de tentativas em memória para ao qual o programa acrescentará os dados restantes. Selecionar o ficheiro no simulador em **Ficheiro->Escreve Memória**.

NOTA: Se sentir dificuldades a desenvolver a alínea b) passe à resolução da alínea c) copiando a subrotina **NPPretos** para **NPBranco**s. Continue para a alínea d) e só após isso volte a pegar na subrotina **NPBranco**s. Assim não correrá o risco de não ter tempo disponível para as alíneas c) e d) por não ter concluído a alínea b).

BOM TRABALHO!

Avaliação

Cotação:

A cotação encontra-se junto de cada uma das alíneas, entre [].

CrITÉRIOS de Correção:

Funcionalidade: 50%

Simplicidade e Modularidade: 10%

Eficiência (serão contabilizados o número de instruções e ciclos de relógio): 10%

Apresentação do código (indentação e comentários): 20%

Relatório (Legibilidade e Justificação dos Resultados e das Opções): 10%

Descontos:

Trabalhos entregues que não estejam em conformidade com as regras de entrega do e-fólio B: até 10%

Código sem comentários, ou apenas com comentários a reflectir o significado da instrução (exemplo MOV R1,R2 ;mover o conteúdo de R2 para R1) : até 50%

Deteção de fraude (total ou parcial): 100%

Trabalhos entregues após a data limite (o recurso de entrega estará aberto até ao início da manhã do dia seguinte, no entanto entregas após a data/hora indicada, têm desconto na nota) : 10%

Regras para entrega do e-fólio B:

Forma de entrega:

Um ficheiro zipado com o nome correspondente ao número de aluno.

O ficheiro zipado deverá por sua vez conter um ficheiro de Assembly do P3 por alínea, cujo nome será o número de aluno mais a letra da alínea ex: 999999a.as.

Deverá ainda ser entregue um pequeno relatório em formato pdf de até 5 páginas A4, com todos os cálculos e todas as opções tomadas na construção dos programas.

Regras relativas à escrita dos programas de Assembly:

Todos os programas devem acabar com a seguinte instrução:

FIM: JMP FIM ;Fim do programa

Não são aceites entregas fora da plataforma Moodle.