

I Semana de Software Livre da USP

Dojo C

MasterMind

25 de abril de 2012

1 Introdução

O MasterMind é um jogo cujo objetivo é descobrir um número de 5 dígitos em 10 tentativas. A cada tentativa, o jogador é informado quantos dígitos estão corretos e quantos dígitos estão na posição certa. Usando lógica, o jogador pode eliminar erros com uma nova tentativa. Mas também ocorre que depois de várias tentativas o seu subconsciente permite ver o número completo em sua mente. Por isso, esse jogo é útil para exercitar os dois lados do seu cérebro.

Parapsicologia à parte, seria interessante saber qual o número mínimo de tentativas suficientes para acertar qualquer número sorteado. Mas essa estória fica para outra vez...

2 O jogo

Inicialmente, um número inteiro de 5 dígitos é sorteado pelo programa, um número entre 0 e 99999. A tarefa do jogador é adivinhar o número que foi sorteado.

A cada passo, o jogador digita um número e o programa revela quantos dígitos estão certos e, dentre estes, quantos estão na posição certa. O jogador ganha se acertar o número sorteado em 10 ou menos tentativas.

Para a contagem dos *dígitos certos* podemos pensar no seguinte. Fazendo uma permutação dos dígitos do número chutado, qual é o maior número de dígitos que eu consigo colocar nas mesmas posições e com o mesmo valor do número sorteado? Esse é o **número de dígitos certos**. Por exemplo, se o número sorteado é 12335 e o número chutado é 32947 a permutação 92347 coloca 2 dígitos na posição certa, e isso é o melhor possível.

Embora os números sejam digitados sempre com 5 ou menos dígitos, todos os dígitos à esquerda, até completar 5 dígitos, são sempre considerados zeros. Num exemplo abaixo, o número sorteado é 10 e o número chutado é 1. Neste caso, consideramos que o número sorteado é 00010 e o número chutado é 00001. Com isso, o número de dígitos certos é 5 e o **número de dígitos na posição certa** é 3, que são os 3 zeros à esquerda.

3 Exemplos

Número sorteado	87345	Número sorteado	22222
Chute	97123	Chute	2
Número de dígitos em posições certas	1	Número de dígitos em posições certas	1
Número de dígitos certos	2	Número de dígitos certos	1

Número sorteado	22345	Número sorteado	11224
Chute	12678	Chute	32178
Número de dígitos em posições certas	1	Número de dígitos em posições certas	0
Número de dígitos certos	1	Número de dígitos certos	2

Número sorteado	10	Número sorteado	2
Chute	1	Chute	22222
Número de dígitos em posições certas	3	Número de dígitos em posições certas	1
Número de dígitos certos	5	Número de dígitos certos	1

4 O quê vocês devem fazer

Vocês devem fazer um programa em C para o jogo MasterMind. Seu programa deve

1. perguntar ao jogador se ele prefere digitar o número a ser adivinhado ou se prefere que o programa sorteie o número;
2. se o jogador deseja que o programa sorteie o número, o seu programa deve pedir ao jogador para digitar um número que será usado como semente para gerar "aleatoriamente" o número a ser adivinhado (veja seção ??); suponha que o número gerado seja armazenado na variável `sorteado`;
3. pedir, no máximo 10 vezes, para o jogador digitar um número `chute`, $0 < \text{chute} < 99999$;
4. informar quantos dígitos de chute aparecem em sorteado na posição certa e quantos dígitos de chute aparecem em sorteado.

No final, imprimir uma mensagem indicando se o jogador adivinhou ou não o número. No caso de o jogador não ter resolvido o jogo, o número sorteado deverá ser informado.

5 Exemplos de execuções do programa

Exemplo 1

Deseja digitar o numero a ser adivinhado? (1 = sim, 0 = nao): 0
Digite uma semente para geracao do numero aleatorio: 1

Tentativa 1: 11111
Numero de digitos em posicoes certas = 1
Numero de digitos certos = 1

Tentativa 2: 22221
Numero de digitos em posicoes certas = 0
Numero de digitos certos = 1

Tentativa 3: 13333
Numero de digitos em posicoes certas = 0
Numero de digitos certos = 1

Tentativa 4: 41444
Numero de digitos em posicoes certas = 0
Numero de digitos certos = 2

Tentativa 5: 54155
Numero de digitos em posicoes certas = 2
Numero de digitos certos = 2

Tentativa 6: 64166
Numero de digitos em posicoes certas = 3
Numero de digitos certos = 3

Tentativa 7: 64177
Numero de digitos em posicoes certas = 3
Numero de digitos certos = 4

Tentativa 8: 74168
Numero de digitos em posicoes certas = 3
Numero de digitos certos = 4

Tentativa 9: 94167
Numero de digitos em posicoes certas = 4
Numero de digitos certos = 4

Tentativa 10: 4167
Numero de digitos em posicoes certas = 5
Numero de digitos certos = 5

Parabens! Voce adivinhou o numero!

Exemplo 2

Deseja digitar o numero a ser adivinhado? (1 = sim, 0 = nao): 0

Digite uma semente para geracao do numero aleatorio: 2

Tentativa 1: 11111

Numero de digitos em posicoes certas = 1

Numero de digitos certos = 1

Tentativa 2: 12222

Numero de digitos em posicoes certas = 0

Numero de digitos certos = 1

Tentativa 3: 31333

Numero de digitos em posicoes certas = 0

Numero de digitos certos = 1

Tentativa 4: 44144

Numero de digitos em posicoes certas = 1

Numero de digitos certos = 2

Tentativa 5: 55514

Numero de digitos em posicoes certas = 2

Numero de digitos certos = 3

Tentativa 6: 65416

Numero de digitos em posicoes certas = 2

Numero de digitos certos = 4

Tentativa 7: 67541

Numero de digitos em posicoes certas = 1

Numero de digitos certos = 4

Tentativa 8: 85641

Numero de digitos em posicoes certas = 0

Numero de digitos certos = 4

Tentativa 9: 96514

Numero de digitos em posicoes certas = 2

Numero de digitos certos = 4

Tentativa 10: 65140

Numero de digitos em posicoes certas = 0

Numero de digitos certos = 5

Voce nao conseguiu adivinhar!

O numero sorteado foi 4516

6 Sorteio de um número pelo programa

Para sortear o número a ser adivinhado usaremos as funções `rand` e `srand` da biblioteca padrão do C (`<stdlib.h>`). O seu programa precisa ter o seguinte trecho de código:

```
#include <stdio.h>
#include <stdlib.h>

#define SIM 1

int main( )
{
    int opcao; /* indica se o numero a ser adivinhado sera sorteado ou nao */
    int sorteado; /* numero a ser adivinhado */
    int semente; /* semente usada para inicializar o gerador de numeros "aleatorios" */

    [ . . . outras declaracoes . . . ]

    /* 1. pergunte se o numero a ser adivinhado deve ser lido */
    printf("Deseja digitar o numero a ser adivinhado? (1 = sim, 0 = nao): ");
    scanf("%d", &opcao);

    /* 2. leia ou sorteie o numero a ser adivinhado */
    if (opcao == SIM)
    {
        printf("Digite o numero a ser adivinhado: ");
        scanf("%d", &sorteado);
    }
    else
    {
        printf("Digite a semente para o gerador de numeros aleatorios: ");
        scanf("%d", &semente);
        srand(semente);
        sorteado = rand()%1000*100 + rand()%100;
    }

    /* Remover a linha seguinte depois de testar o programa */
    printf("Numero sorteado: %d\n", sorteado);

    [ . . . resto do programa . . . ]

    return 0;
}
```

Para uma mesma semente, o número sorteado é sempre o mesmo.