



Projeto 1

Profa. Simone Aires

Crie um programa que implemente o Jogo da Senha (por exemplo: <http://www.usinadejogos.com.br/senha.html>). O jogo da senha gera um número aleatório e deve permitir que o usuário adivinhe este número. Considere que os elementos possíveis para este programa são os algarismos 0, 1, 2, 3, ... 9. A cada tentativa do usuário, o jogo deverá lhe informar quantos elementos estão certos na posição certa e quantos estão certos mas na posição errada. Por exemplo:

O jogo gera o número: **5219**. E o usuário joga:

Tentativa	Número	Resposta
1	2381	0 2
2	4298	1 1
3	8267	1 0
4	9215	2 2
5	1295	1 3
6	5219	4 0 - Você acertou em 6 tentativas.

O jogo deve permitir 3 níveis de dificuldade (fácil, difícil e teste) – o nível fácil usa senhas de apenas 3 dígitos sem repetição, o nível difícil usa senhas de 4 dígitos sem repetição. O modo teste é igual ao difícil, mas a senha aparece para o usuário. O número de tentativas máximo é um parâmetro a ser solicitado do usuário. A interface do jogo pode ser completamente textual, mas você deve utilizar o TAD criado para implementar o jogo.

O seu programa deverá implementar as funções `ler_senha`, `sorteia_senha` e `compara_senha`.

- A função `ler_senha` deve receber como parâmetro a quantidade de números que irão compor a senha e um vetor previamente alocado dinamicamente para armazenar a quantidade exata de números gerados com base no nível de dificuldade selecionado. A função deve pedir para o usuário digitar os números e armazená-los no vetor, garantindo que somente números dentro do intervalo de 01 a 9 sejam digitados. A função deve seguir o seguinte protótipo:

```
void ler_senha(int *aposta, int n);
```

- A função `sorteia_valores` deve receber como parâmetro a quantidade de números que serão sorteados e um vetor previamente alocado dinamicamente para armazenar a quantidade exata de números sorteados. A função deve sortear aleatoriamente os números (entre 01 e 9) e armazená-los no vetor. A função deve seguir o seguinte protótipo:

```
void sorteia_valores(int *sorteio, int n);
```



- A função `compara_senha` deve receber como parâmetro o vetor com os números do usuário, o vetor com os números sorteados (`senha`), juntamente com os seus respectivos tamanhos (`na` e `ns`) e um ponteiro para uma variável inteira (`qtdacertos`), onde deve ser armazenada a quantidade de acertos. A função deve retornar o ponteiro para um vetor alocado dinamicamente contendo as posições dos números que o usuário acertou. A função deve seguir o seguinte protótipo:

```
void compara_senha(int *senha, int *senhausuario, int *qtdacertospc, int *qtdacertospe, int dificuldade);
```

Em seguida, crie a função principal do programa utilizando as funções criadas anteriormente para implementar o jogo. Os vetores devem ser alocados dinamicamente e a memória alocada deve ser liberada quando ela não for mais utilizada. Lembre-se que um usuário poderá jogar quantas partidas desejar, em qualquer um dos níveis.

Para sortear números aleatórios utilize a função `rand` da biblioteca `stdlib.h`. A função `rand` retorna um número aleatório em um determinado intervalo. Exemplo:

```
x = rand() % 10; /* x vai receber um valor entre 0 e 10 */
```

Para garantir que novos números aleatórios sejam sorteados em cada execução do programa é necessário executar a função `srand` antes de sortear os números. Exemplo:

```
srand(time(NULL));
```

Para poder utilizar essas funções é necessário incluir no programa as bibliotecas `stdlib.h` e `time.h`. Exemplo de programa para sortear um número aleatório:

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

```
int main(void)
{
    int x; srand(time(NULL));
    x = rand() % 60; /* x vai receber um valor entre 0 e 60 */
    printf("%d", x); return 0;
}
```