

# MAC0110 – Introdução à Computação

BCC 2022

Primeiro Exercício-Programa – Data de entrega: até 27/04/2022

Diz a sabedoria popular que **É brincando que se aprende!**

Aqui vamos inverter e praticar o **É aprendendo que se brinca!**

Mais especificamente, neste EP iremos implementar um programa em linguagem C que fará a simulação de JOGOS DE FORCA.

Leia o enunciado com cuidado. Lá no final há instruções IMPORTANTES!

## O que o programa deve fazer

O jogo de forca é normalmente jogado com palavras e letras. Neste EP faremos uma versão com números e dígitos. A cada passo do jogo, o jogador deverá chutar um dígito, até acertar o número oculto.

O programa deverá permitir que o usuário jogue o jogo tantas vezes quanto queira. Cada jogo estará associado a um número inteiro de 000 a 999, que é o número que o jogador tentará adivinhar. Esse número deverá ser sorteado pelo programa (via geração de números pseudo-aleatórios – veja mais detalhes na seção final deste enunciado). Antes de cada jogo, o programa deve apresentar as seguintes informações e o menu de opções (o exemplo abaixo corresponde à impressão antes do primeiro jogo):

```
[ depuração = DESLIGADO / jogos jogados = 0 ]
1. Continuar a jogar
2. Alternar modo depuração e continuar a jogar
3. Parar
Escolha uma opção:
```

A opção 2 permite que o modo de depuração LIGADO ou DESLIGADO, seja alternado. Inicialmente, o modo depuração deve estar DESLIGADO. A escolha da opção 2 alterna imediatamente o modo, e o jogo segue de acordo com o novo modo de depuração. Se o modo depuração estiver LIGADO, o número oculto deverá ser revelado antes do usuário começar a fazer os chutes (embora isso não faça sentido em um Jogo de Forca, no EP ele será útil para verificarmos se o programa está funcionando como deveria).

Em seguida, o usuário poderá chutar dígitos, um de cada vez, para tentar descobrir o número oculto. A cada passo, o programa deve ler o chute do usuário e deve revelar onde o dígito aparece no número, se este for o caso. O usuário pode fazer até 6 tentativas por jogo, e ganhará o jogo se acertar os 3 dígitos do número (note que pode haver dígitos repetidos) e perderá em caso contrário. A cada jogo o programa deve informar se o jogador VENCEU ou PERDEU o jogo.

Quando o usuário escolhe a opção 3 (Parar), o programa deve imprimir quantas vezes o jogador jogou e quantas vezes ele ganhou, e então terminar a simulação.

## Exemplo de execução do programa

Bem-vindo ao JOGO da FORCA !!

~~~~~

```
[ depuração = DESLIGADO / jogos jogados = 0 ]
1. Continuar a jogar
2. Alternar modo depuração e continuar a jogar
3. Parar
Escolha uma opção: 1
```

Restam 6 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 0  
Você já acertou os seguintes dígitos: \_ \_ \_

Restam 5 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 8  
Você já acertou os seguintes dígitos: \_ 8 \_

Restam 4 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 2  
Você já acertou os seguintes dígitos: \_ 8 \_

Restam 3 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 1  
Você já acertou os seguintes dígitos: \_ 8 \_

Restam 2 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 7  
Você já acertou os seguintes dígitos: \_ 8 \_

Resta uma tentativa!  
Sua próxima tentativa (dígito de 0 a 9): 9  
Você já acertou os seguintes dígitos: \_ 8 \_

Você PERDEU!

```
[ depuração = DESLIGADO / jogos jogados = 1 ]
1. Continuar a jogar
2. Alternar modo depuração e continuar a jogar
3. Parar
Escolha uma opção: 2
```

Número a ser adivinhado: 8 8 6  
Restam 6 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 5  
Você já acertou os seguintes dígitos: \_ \_ \_

Restam 5 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 6  
Você já acertou os seguintes dígitos: \_ \_ 6

Restam 4 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 9  
Você já acertou os seguintes dígitos: \_ \_ 6

Restam 3 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 3  
Você já acertou os seguintes dígitos: \_ \_ 6

Restam 2 tentativas

Sua próxima tentativa (dígito de 0 a 9): 2  
Você já acertou os seguintes dígitos: \_ \_ 6

Resta uma tentativa!  
Sua próxima tentativa (dígito de 0 a 9): 8  
Você já acertou os seguintes dígitos: 8 8 6

Você VENCEU!

[ depuração = LIGADO / jogos jogados = 2 ]  
1. Continuar a jogar  
2. Alternar modo depuração e continuar a jogar  
3. Parar  
Escolha uma opção: 1

Número a ser adivinhado: 7 7 7  
Restam 6 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 7  
Você já acertou os seguintes dígitos: 7 7 7

Você VENCEU!

[ depuração = LIGADO / jogos jogados = 3 ]  
1. Continuar a jogar  
2. Alternar modo depuração e continuar a jogar  
3. Parar  
Escolha uma opção: 2

Restam 6 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 5  
Você já acertou os seguintes dígitos: \_ \_ 5

Restam 5 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 3  
Você já acertou os seguintes dígitos: \_ \_ 5

Restam 4 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 1  
Você já acertou os seguintes dígitos: \_ 1 5

Restam 3 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 0  
Você já acertou os seguintes dígitos: \_ 1 5

Restam 2 tentativas  
Sua próxima tentativa (dígito de 0 a 9): 9  
Você já acertou os seguintes dígitos: 9 1 5

Você VENCEU!

[ depuração = DESLIGADO / jogos jogados = 4 ]  
1. Continuar a jogar  
2. Alternar modo depuração e continuar a jogar  
3. Parar  
Escolha uma opção: 3

Você jogou 4 jogos e venceu 3 vezes!

## Algumas informações adicionais

**Sorteio de um número de 000 a 999:** Para tanto, inclua além de `stdio.h`, o `stdlib.h`. Assim as primeiras linhas do seu programa serão:

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

int main() {
```

Para sortear um número entre 000 e 999, basta fazer do seguinte modo:

```
num = rand() % 1000;
```

Esta é uma forma simplificada para a geração de números aleatórios, que para os propósitos deste EP está mais do que suficiente.

Há várias discussões sobre cuidados a serem tomados na geração de números aleatórios. Veja, por exemplo, <https://stackoverflow.com/questions/822323/how-to-generate-a-random-int-in-c>.

**Consistência de dados:** Não há necessidade de fazer consistência dos dados de entrada. Isto significa que não é preciso verificar se os dados digitados pelo usuário são válidos. O programa será testado somente com valores válidos.

**Comandos permitidos:** Utilize apenas variáveis do tipo `int`. utilize apenas os comandos da linguagem C que estamos usando em sala de aula.

**Organização do código:** Códigos confusos poderão sofrer redução de nota. Preste atenção na indentação, use variáveis com nomes que fazem sentido, adicione comentários para as partes não tão óbvias se pertinente (não comente o óbvio).

**Saída do programa:** Procure fazer o mais próximo possível do exemplo dado acima.

**Compilação:** Compile o seu programa antes de entregar. Programas com erro de compilação não serão corrigidos. Teste o programa antes de entregar. **Você deve entregar o código-fonte (isto é, o arquivo com o programa escrito em linguagem C).** Para compilar, por exemplo um arquivo com nome `ep1.c`, use o comando

```
gcc -ansi -Wall ep1.c -o ep1
```

**Dúvidas:** Recomenda-se que as dúvidas sejam postadas no **Fórum de discussões** no *edisciplinas*. Podem também ser discutidas/esclarecidas em sala de aula ou com os monitores.

— Discussões sobre o problema e ideias com os seus colegas são permitidos, mas **a feitura do programa dever ser um trabalho individual.** —