



# Instituto Federal Minas Gerais – Campus Formiga

Ciência da Computação - 1º Semestre 2021

Disciplina: Algoritmos I

Prova – 30 Pontos

INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
Campus Formiga-MG

Nome: \_\_\_\_\_ Data: **14/06/2021**

## → INSTRUÇÕES. LEIA ATENTAMENTE ANTES DE INICIAR A PROVA:

1. A prova deve ser feita utilizando a linguagem C.
2. Todas as questões devem estar de acordo com as boas práticas de programação (comentários, endentação, etc).
3. A prova deve ser feita de forma INDIVIDUAL. O aluno pego utilizando qualquer meio ilícito terá sua nota ZERADA.
4. A interpretação faz parte da prova, por isso só serão respondidas dúvidas a respeito de falha de impressão.
5. Crie uma pasta com o seu nome\_sobrenome (Ex.: Manoel\_Junior) e grave todos os arquivos dentro dela.
6. Salve cada questão com o seguinte formato questaoX.c, onde X é o número da questão. Ex.: questao1.c
7. **Compacte a pasta criada com suas resoluções e envie para o local correto no sistema acadêmico.**
8. O tempo de prova inclui a criação e gravação dos arquivos e pastas.
9. **Caso seja necessário, escreva suas considerações NAS FOLHAS DE PROVA, devidamente assinadas.**

### Questão 1 (50%)

Existe um jogo de cartas onde o usuário joga contra o computador (o seu programa) e tenta vencê-lo. O objetivo do jogo é obter o valor mais próximo de 26, sem ultrapassá-lo. Cada carta do baralho (que vai de A a K) possui o seu respectivo valor. Assim, a carta A vale 1, a carta 2 vale 2, a carta 3 vale 3, ..., a carta 10 vale 10, carta J vale 11, a carta Q vale 12 e finalmente a carta K vale 13, independente do naipe. Ao iniciar o jogo, o computador sorteia **duas** cartas para ele, **soma esses valores** mas **não** os mostra para o usuário. Por exemplo, se ele sair com uma carta 2 e com uma carta 7, seu valor será 9.

Em seguida, é a vez do usuário. Ele pode pedir que o programa sorteie até 5 cartas para ele ou pode resolver que as cartas que tem já são suficientes para ganhar o jogo. Por exemplo, ao pedir a primeira carta o usuário sai com um rei (K) que vale 13. Ao pedir a segunda, ele tira outro rei (K), somando 26. Como já tem o número “alvo” (ou seja, o número 26), ele pode escolher parar. Em outro caso, caso tenha apenas cartas pequenas e julgue estar longe do “alvo”, o usuário pode escolher até 5 cartas para compor o seu número. Quando escolhe parar ou quando recebe as 5 cartas, o jogo termina. Porém, se o valor somado do jogador ultrapassar 26, ele perde o jogo imediatamente. Caso isso não aconteça, ganha o jogo quem tiver o valor mais próximo de 26. Faça um programa que implemente esse jogo. **NÃO É NECESSÁRIO DESENHAR NADA NA TELA!**

### Questão 2 (50%)

Outro jogo bastante conhecido é o campo minado. Ao iniciar este jogo, o jogador recebe um tabuleiro com diversos quadrinhos. Debaixo de cada quadrinho pode ou não existir uma mina. A missão do jogador é abrir todos os quadrinhos que não possuem minas. Se ele clica em um quadrinho que tem uma mina, o jogo termina. Veja o exemplo de um tabuleiro ao lado (O EXEMPLO É APENAS ILUSTRATIVO):



Agora que você já conhece o jogo campo minado, faça um programa que:

1. Considere um tabuleiro com dimensões 5x5 (ou seja, 25 posições possíveis).
2. Posicione 5 minas ao iniciar, cada uma ocupando apenas uma posição (um quadrinho). Desconsidere posições repetidas.
3. Permita que o usuário abra 8 quadrinhos. Seu programa deve informar se o jogador abriu um quadrinho com ou sem mina. Se for uma mina, seu programa deve informar o ocorrido e encerrar.
4. Informe ao jogador que ele ganhou a partida, caso ele não tenha revelado nenhuma mina após ter aberto os 8 quadrinhos.
5. **NÃO É NECESSÁRIO DESENHAR NADA NA TELA!**

### Cola Oficial

Em ambas as questões, será necessário gerar números “aleatórios”. Para fazer isso em C, utilize as seguintes funções:

```
#include <time.h>          /* insira este comando antes do main */
srand(time(NULL));         /* comando para setar a semente da máquina pseudo-aleatória, usado uma
                             única vez*/
int numero=rand()%10; /* gera um número aleatório entre 0 e 9. Note que estou usando
                       um operador de mod. Se quiser gerar um número “aleatório”
                       entre 1 e 10 basta somar 1.*/
```

Considerações e rascunho do aluno no verso da folha: