

LISTA DE EXERCÍCIOS C/C++/ED

PROF. RAMON

1. Considere que você tem as seguintes variáveis em C:

```
int resposta = 42;
double pi = 3.1415926536;
char nome[50] = "Letícia";
```

Agora, escreva um comando para imprimi-las na tela, da seguinte forma:

```
resposta = 42
pi = 3.1415926535
nome = Letícia
```

2. Considere as mesmas variáveis em um programa em C++. Qual seria o comando para imprimi-las na tela?
Obs.: pesquise como se faz a formatação de números reais no C++ para que o pi apareça com 10 casas decimais.
3. Faça em C: leia dois números reais e mostre a soma entre eles.
4. Faça em C++: leia dois números reais e mostre a soma entre eles.
5. Faça em C um programa que mostra o seguinte texto:

```
*   ***   ***   ***
*   *       *   *
*   ***   *   ***
*   *       *   *
*** ***   *   ***
```

6. Faça em C++ um programa que mostra o mesmo texto do exercício anterior.
7. Faça em C um programa que mostra o seguinte texto:

```
      X
     XXX
    XXXXX
   XXXXXXX
  XXXXXXXXX
 XXXXXXXXXXX
XXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX
      XXX
      XXX
     XXXX
```

8. Faça em C++ um programa que mostra o mesmo texto do exercício anterior.

9. Faça em C++ um programa que mostra o mesmo texto do exercício anterior, mas que pergunte a altura do pinheiro antes. A altura deve ser maior ou igual a 3 e menor ou igual a 10.

Exemplo:

```
Entre com a altura do pinheiro (entre 3 e 10): 2
Tamanho inválido...
Entre com a altura do pinheiro (entre 3 e 10): 2
Tamanho inválido...
Entre com a altura do pinheiro (entre 3 e 10): 5
```

```

  X
 XXX
XXXXX
XXXXXXX
XXXXXXXXX
  XXX
  XXX
  XXXX
```

10. Faça em C++ um programa que sorteia um número entre 1 e 10 e peça para o usuário adivinhar que número é.

Exemplo 1:

```
Pensei em um número entre 1 e 10. Adivinhe qual é...
Tentativa: 11
É um número entre 1 e 10...
Tentativa: 4
Errou. O número sorteado era o 6.
```

Exemplo 2:

```
Pensei em um número entre 1 e 10. Adivinhe qual é...
Tentativa: 7
Acertou!
```

11. Melhore o programa anterior para que o usuário possa tentar várias vezes.
12. Melhore ainda mais o programa anterior para que o computador dê dicas – se o número sorteado é maior ou menor que o número da última tentativa.

Exemplo:

```
Pensei em um número entre 1 e 10. Adivinhe qual é...
Tentativa: 8
É um número menor...
Tentativa: 3
É um número maior...
Tentativa: 4
Acertou!
```

13. Melhore ainda mais o programa anterior para que o computador mostre, no final, quantas tentativas foram usadas para acertar o número.

Exemplo:

```
Pensei em um número entre 1 e 10. Adivinhe qual é...
Tentativa: 8
É um número menor...
Tentativa: 3
É um número maior...
Tentativa: 4
Acertou em 3 tentativas!
```

14. Melhore ainda mais o programa anterior para que o usuário só possa tentar 3 vezes.

Exemplo:

```
Pensei em um número entre 1 e 10. Adivinhe qual é...
Tentativa 1: 8
É um número menor...
Tentativa 2: 3
É um número maior...
Tentativa 3: 4
Errou. O número sorteado era o 6.
```

15. Faça o mesmo jogo, mas de forma invertida, ou seja, você pensa num número entre 1 e 10 e o computador tem que acertar em, no máximo, 3 tentativas. Você vai dando as dicas...
Observação: qual seria uma boa estratégia para o computador adivinhar o número?

Exemplo:

```
Pense em um número entre 1 e 10. Pensou? (s/n): s
É o 5? (s/n): n
É maior ou menor? (</>): >
É o 7? (s/n): n
É maior ou menor? (</>): <
É o 7? (s/n): s
Acertei!!!
```

16. O dígito verificador do PIS/PASEP é calculado através da seguinte regra: o número é composto por dez dígitos mais um dígito verificador. Multiplique os números, da esquerda para a direita, respectivamente por 3 2 9 8 7 6 5 4 3 2. Some os resultados das multiplicações; calcule o resto da divisão da soma por 11 e subtraia o resultado de 11. Se o resultado for 10 o dígito é zero, caso contrário o dígito é o próprio resultado.

Por exemplo, para o número 1701209041-1, o cálculo seria:

$$1 \times 3 + 7 \times 2 + 0 \times 9 + 1 \times 8 + 2 \times 7 + 0 \times 6 + 9 \times 5 + 0 \times 4 + 4 \times 3 + 1 \times 2 = 98.$$

O resto da divisão de 98 por 11 é 10.

Como $11 - 10 = 1$, o dígito é 1.

Escreva um programa em C++ que lê um número de PIS/PASEP e mostra o dígito verificador correspondente. Para testá-lo, você pode usar também o número 1010861269-1.

17. Melhor o programa anterior de forma que o número do PIS/PASEP seja representado por um Tipo Abstrato de Dado.
18. Um vulcão acaba de entrar em erupção, provocando uma nuvem de cinzas que se alastra impedindo a circulação aérea. O governo está muito preocupado e deseja saber quando que a nuvem de cinzas irá atingir todos os aeroportos do país.

Está disponível um mapa detalhando a situação atual. O mapa é retangular, dividido em pequenos quadrados. Neste mapa existem três tipos de quadrados: nuvem (indicando que esta região do mapa já está coberto por nuvens), aeroportos (indicando a localização de um aeroporto) e todas as outras (indicando locais onde a nuvem ainda não chegou).

A cada dia, a nuvem expande-se um quadrado na horizontal e um quadrado na vertical. Ou seja, ao fim de cada dia, todos os quadrados adjacentes (vertical ou horizontalmente) a uma nuvem, também passam a conter nuvens. Por exemplo:

. . * . . . * *	. * * * . * * *	* * * * * * * *
. * *	* * * * . . * *	* * * * * * * *
* * * . A . . A	* * * * A . . A	* * * * * . * *
. *	* * *	* * * *
. * A .	* * * . . . A .	* * * * . . A .
. . . A * . A	* * * A
. *
Dia 1	Dia 2	Dia 3

Para preparar os planos de contingência, o governo necessita saber: quantos dias demorará para ao menos um aeroporto ficar coberto pelas nuvens e daqui quantos dias todos os aeroportos estarão cobertos pelas nuvens.

Dados um quadriculado com L linhas e C colunas, além da indicação inicial das nuvens e dos aeroportos, desenvolva uma programa em C++ que informe o número de dias até um primeiro aeroporto ficar debaixo da nuvem de cinzas e o número de dias até que todos os aeroportos fiquem cobertos pelas cinzas.

19. Faça um programa em C++ que lê o nome e as notas de duas provas de um aluno e que mostra a seguinte mensagem: "O aluno Nome do Aluno teve média Média do Aluno".
20. Faça um programa em C++ que lê o nome e as notas de duas provas de um aluno e que mostra a seguinte mensagem: "O aluno Nome do Aluno **passou** com média Média do Aluno" caso ele tenha passado ou "O aluno Nome do Aluno **não passou** com média Média do Aluno" caso ele não tenha atingido a média.
21. Faça um programa em C++ que lê o nome e as notas de duas provas de **vinte** alunos e que, depois, mostra dois relatórios: um de alunos aprovados e outro de alunos reprovados.
22. Melhore o programa anterior de forma que o usuário possa entrar, primeiramente, com o número de alunos que existe na sala.
23. Melhore o programa anterior de forma que o usuário possa entrar com os dados dos alunos sem se preocupar com quantos são. Para indicar que o não existem mais alunos a serem lidos, o usuário entra com um nome vazio.

24. Faça um programa em C++ capaz de gerenciar um cadastro de alunos (matrícula, nome, nota da P1 e nota da P2):

```
1-Inserir aluno
2-Listar alunos
3-Consultar aluno
4-Atualizar aluno
5-Remover aluno
6-Listar alunos aprovados
7-Listar alunos reprovados
```

Use uma lista genérica implementada em vetor para armazenar os alunos.

25. Melhore o programa anterior de forma que a lista de alunos fique sempre em ordem alfabética.

26. Escreva um programa em C++ que simule o controle de uma pista de decolagem de aviões em um aeroporto. Neste programa, o usuário deve ser capaz de realizar as seguintes tarefas:

```
1-Mostrar a quantidade de aviões aguardando na fila de decolagem
2-Autorizar a decolagem do primeiro avião na fila
3-Adicionar um avião na fila
4-Listar todos os aviões na fila
5-Mostrar os dados do primeiro avião da fila
```

Considere que os aviões possuem um nome e um número inteiro como identificador. Adicione outros dados conforme achar necessário

Use uma fila genérica implementada em vetor para armazenar os aviões.

27. Altere o programa anterior para usar uma fila **circular** para armazenar os aviões.

28. Faça um programa em C++ que permita o usuário brincar com as Torres de Hanoi de 3 discos.

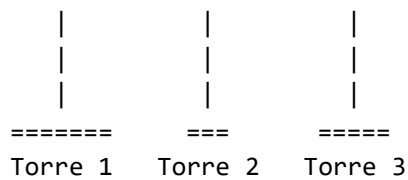
Exemplo:

```
  |      |      |
  ==
  =====
  =====
Torre 1  Torre 2  Torre 3
```

Mover da torre: 1
Para a torre: 2

```
  |      |      |
  |      |      |
  =====
  =====  ==
Torre 1  Torre 2  Torre 3
```

Mover da torre: 1
Para a torre: 2
Movimento inválido (não pode colocar um disco maior em cima de um menor)
Mover da torre: 1
Para a torre: 3



Use uma pilha genérica implementada em vetor para representar as torres e armazenar os discos.

29. Faça um programa em C++ que recebe uma string formada por letras e dígitos e mostre uma nova string na qual as letras são mantidas na sequência original e os dígitos são colocados na ordem inversa. Exemplos:

A 1 E 5 T 7 W 8 G → A E T W G 8 7 5 1
 3 C 9 H 4 Q 6 → C H Q 6 4 9 3

Como mostram os exemplos, as letras devem ser mostradas primeiro, seguidas dos dígitos.

30. Faça um programa em C++ que implemente uma calculadora que funcione parecido com uma HP12c, ou seja, que usa a notação polonesa reversa.

Exemplo:

Visor: 0
 > 10
 Visor: 10
 > 20
 Visor: 20
 > +
 Visor: 30
 > 40
 Visor: 40
 > *
 Visor: 1200
 > L
 Visor: 0
 > S

Permita as operações básicas (+, -, *, /) e as seguintes funcionalidades:

- L = limpar: esvazia a pilha (situação inicial)
- S = sair