

# Cheat Sheet dia 2

## Comandos condicionais

```

if (teste lógico) {
    //comandos
} else if ( teste lógico) {
    //comandos
} else {
    //comandos
}

switch (variável ) {
    case valor1 :
        [sequencia de comandos];
        break;
    case valor2:
        [sequencia de comandos];
        break;
    ..
    case valorN:
        [sequencia de comandos];
        break;
    default:
        [sequencia de comandos];
}

```

## Comandos de repetição

```

while (teste lógico) {
    //comandos;
}

do {
    //comandos;
} while (teste lógico)

for (i = 0; i < 10; i++) {
    //comandos;
}

```

## Tabelas verdade para os operadores lógicos

a	b	a && b
F	F	F
F	V	F
V	F	F
V	V	V

a	b	a    b
F	F	F
F	V	V
V	F	V
V	V	V

## 1. Aquecendo 1

Vamos treinar um pouco a lógica. Utilize o comando IF.

1. Leia do usuário 2 variáveis do tipo inteiro
2. Crie testes lógicos que verifiquem as perguntas abaixo e imprimam na tela a resposta para uma delas.
  - a.  $a$  é igual a  $b$ ?
  - b.  $a * b$  é menor do que 100?
  - c. O triplo de  $a$  é maior do que  $b$ ?
  - d.  $a$  ou  $b$  é maior do que 0?
  - e. Os dois números são simultaneamente negativos?
  - f. Algum dos dois números é ímpar?
  - g. Imprima o menor entre os dois números

### Entrada

7 10

### Saída

Nao

Sim

Sim

Sim

Nao

Sim

7

## 2. Aquecendo 2

Faça um programa que receba três números inteiros e:

- a. Imprima se há números repetidos
- b. Caso **não** haja números repetidos, imprima o maior deles
- c. Caso haja números repetidos, imprima o menor deles
- d. Imprima quais são os números pares (se houver)
- e. Imprima quais são os números ímpares (se houver)
- f. Imprima quais são os números positivos (se houver)
- g. Imprima quais são os números negativos (se houver)

**Exemplos**

<b>Entrada</b>	<b>Entrada</b>
10 20 30	-5 7 3
<b>Saída</b>	<b>Saída</b>
nao	nao
30	7
10 20 30	-5 7 3
10 20 30	7 3
	-5

### 3. Quantos Impar?

Faça um programa que receba 5 números inteiros e conte quantos são impar. Precisaremos de uma variável auxiliar que guarde essa quantidade. Pode chamá-la de “count”, por exemplo.

- a. Essa variável deverá ser inicializada com 0 para evitar o problema de lixo de memória e incrementar de uma unidade a cada vez que identificar um número par.
- b. Você pode incrementar fazendo:
  - i. `count = count + 1;`
  - ii. `count += 1;`
  - iii. `count++;` (essa é a forma mais comum)

**Exemplos**

<b>Entrada</b>	<b>Entrada</b>
1 2 3 4 5	0 -5 22 35 -74
<b>Saída</b>	<b>Saída</b>
Quantos sao par: 2	Quantos sao impar: 2

### 4. Empréstimos

Um banco quer garantir que os clientes terão condições de pagar pelo empréstimos que estão solicitando. Para isso, define a seguinte regra: a prestação do empréstimo não pode ser maior do que 20% do salário do cliente.

1. Faça um programa que leia o salário do cliente e o valor da prestação.
2. Se qualquer um dos valores lidos for negativo, imprima “Valores invalidos” e encerra o programa
3. Se a prestação for maior que 20% do salário imprima: “Emprestimo nao concedido”, caso contrário imprima: “Emprestimo concedido”.

**Exemplos**

<b>Entrada</b>	<b>Entrada</b>	<b>Entrada</b>
1000 350	2500 500	2500 -200
<b>Saída</b>	<b>Saída</b>	<b>Saída</b>
Emprestimo nao concedido	Emprestimo concedido	Valores invalidos

## 5. Peso ideal

Faça um programa que receba o sexo e a altura de uma pessoa e imprima seu peso ideal conforme as fórmulas abaixo. Para receber o sexo, utilize os caracteres 'h' ou 'm'. Imprima com duas casas decimais

- Homem:  $(72.7 * \text{altura}) - 58$
- Mulher:  $(62.1 * \text{altura}) - 44.7$

**Exemplos**

<b>Entrada</b>	<b>Entrada</b>
h 1.70	m 1.65
<b>Saída</b>	<b>Saída</b>
65.59	57.77

## 6. Anastácia e suas férias

Anastácia conseguiu passar na matéria que estava com dificuldades, e agora está de férias. Ela precisa, agora, saber quantos dias faltam para suas férias acabarem! Ajude Anastácia fazendo um programa que receba:

- uma data inicial (dois inteiros, representando dia e mês do ano)
- uma data final (dois inteiros, representando dia e mês do ano) E imprima a diferença entre essas datas em dias.

Note que agora não há restrições: a data inicial pode ser anterior à final (ambas no mesmo ano) ou posterior à final (em anos diferentes). Se necessário, assuma que 1 mês tem sempre 30 dias, 1 ano tem sempre 360 dias e a diferença entre as datas é sempre menor que 1 ano.

**Exemplos**

<b>Entrada</b>	<b>Entrada</b>	<b>Entrada</b>	<b>Entrada</b>	<b>Entrada</b>
10 07	10 08	10 08	12 08	12 08
10 08	10 07	10 07	10 08	12 08
<b>Saída</b>	<b>Saída</b>	<b>Saída</b>	<b>Saída</b>	<b>Saída</b>
30	330	330	358	0

## Exercícios do URI

Problemas nº 1035, 1037, 1049, 1038, 1042, 1050

===== Comandos de Repetição =====

## 7. Aquecendo 3

Os testes lógicos são executados pelo processador, que responde para o programa com “1” caso o teste seja verdadeiro e com “0” caso o teste seja falso. O teste lógico passado para os comandos condicionais e de repetição só precisam verificar se o retorno é 0 ou diferente de zero. Se o número entre parênteses for diferente de zero, o teste é interpretado como “verdadeiro”.

1. Escreva um programa que escreva a mensagem “Eu não paro de imprimir!” infinitas vezes, uma linha de cada vez. (não faça isso sem antes descobrir como se interrompe um programa)
2. Isso também funciona para variáveis. Crie uma variável ‘aux’ do tipo int, atribua o valor -5 e utilize essa variável para realizar o teste lógico.  

```
while (aux) {  
    printf(“Eu não paro de imprimir!\n”);  
}
```
3. Agora coloque um ! antes de aux. O que aconteceu? Por quê? 

```
while (!aux) {  
    printf(“Eu não paro de imprimir!\n”);  
}
```

## 8. Contando de 1 a n

1. Faça um programa que leia um número inteiro e imprima todos os números de 1 até esse inteiro. Faça isso utilizando o comando “for”.
2. Impressão: entre cada número, deve ter apenas um espaço. Mas ao final, deve ser comandado o “\n”.

### Exemplo 1

**Entrada**

```
7
```

```
Saída
```

```
1  2 3 4 5 6 7
```

## 9.Tabuada

1. Faça um programa que leia um número inteiro e imprima a tabuada de 1 até 20 desse número
2. A impressão deve ser a mensagem conforme exemplo.

### Exemplo 1

```
Entrada
```

```
4
```

```
Saída
```

```
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
4 x 11 = 44
4 x 12 = 48
4 x 13 = 52
4 x 14 = 56
4 x 15 = 60
4 x 16 = 64
4 x 17 = 68
4 x 18 = 72
4 x 19 = 76
4 x 20 = 80
```

## 10. Jogo de adivinhação

Pedro aprendeu a programar e resolveu criar um jogo de adivinhação. Tudo o que o jogador precisa fazer é adivinhar o número exato que Pedro pensou. Escreva um programa que recebe repetidamente um número inteiro do usuário e só para de receber números caso o usuário digite o número 5.

## 11. Fatorial

Faça um programa que, dado um número inteiro, imprima o fatorial desse número. Lembrando: fatorial é multiplicar por todos os números que antecedem o número em questão. Exemplo:

$$10! = 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 3628800$$

Teste seu programa para os valores 3, 5, 10 e 17. O que acontece com o 17?

### Exemplo 1

<i>Entrada</i>	<i>Entrada</i>	<i>Entrada</i>	<i>Entrada</i>
3	5	10	17
<i>Saída</i>	<i>Saída</i>	<i>Saída</i>	<i>Saída</i>
6	120	3628800	o que aconteceu?

## 12. 2 ou 1

Um jogo muito popular para fazer decisões entre um grupo de 3 pessoas é o “Dois ou Um” (também chamado de “Zerinho ou Um”). O jogo consiste em três pessoas escolherem um número, que pode ser 1 ou 2. Sendo três pessoas, uma das pessoas deve tirar o número diferente das outras duas (caso contrário, o jogo se repete).

Faça um programa que analise partidas de Dois ou Um, da seguinte forma:

- Primeiro, receba o valor de cada um dos três jogadores (1 ou 2 cada)
- Depois, imprima o jogador que perdeu (jogador 1, 2 ou 3, em ordem), ou seja, quem tirou um número diferente dos demais. Encerre o programa.
- Caso todos os números sejam iguais, imprima uma mensagem do tipo “Tente Novamente” e continue até que o jogo seja válido

### Exemplo 1

<i>Entrada</i>	<i>Entrada</i>	<i>Entrada</i>
1 1 1	1 2 2	2 2 2
1 2 1		1 1 1

		2 2 2 2 2 2 2 2 1
<b>Saída</b>	<b>Saída</b>	<b>Saída</b>
Tente novamente Jogador 2	Jogador 1	Tente novamente Tente novamente Tente novamente Tente novamente Jogador 3

## 13. Irmãs competitivas

Amanda e Beatriz são irmãs que gostam muito de competir, até nas coisas mais bobas. Um dia, resolveram jogar um jogo no qual ganha a rodada quem escreve o maior número numa folha de papel, e ganha o jogo quem ganhar mais rodadas. Ajude Amanda e Beatriz a competir fazendo um programa que calcule a pontuação de cada uma até que as rodadas acabem e imprima a pontuação final das duas.

O seu programa deve:

- Receber um inteiro  $N$  que significa o número de rodadas
- Para cada rodada, receber dois inteiros, que significam o número escrito por Amanda e por Beatriz
- Ao final de receber todos os valores das rodadas, imprimir a pontuação final de cada uma

### Exemplo 1

<b>Entrada</b>	<b>Entrada</b>
3 23 43 12 22 3 3	5 6 3 13 13 63 33 2 56 -45 -102
<b>Saída</b>	<b>Saída</b>
Ana: 0 Beatriz: 2	Ana: 3 Beatriz: 1



## 14. Bháskara!

Faça um programa que, dados três coeficientes (A, B e C), calcule as raízes do polinômio de segundo grau com esses coeficientes usando a fórmula de Bháskara (oi Google).

- dica: use a biblioteca math.h com a função sqrt( )
- Verifique se há duas, uma ou nenhuma raiz real
- Imprima as raízes com três casas decimais

<i>Entrada</i>	<i>Entrada</i>	<i>Entrada</i>	<i>Entrada</i>
1 2 3	2 -8 0	-2 -25 3	1 0 0
<i>Saída</i>	<i>Saída</i>	<i>Saída</i>	<i>Saída</i>
Nao ha raizes reais	2 raizes reais: 0.000 e 4.000	2 raizes reais: -12.619 e 0.119	1 raiz real: 0.000

## 15. Raiz quadrada

Faça um programa que leia um inteiro N e imprima todos os números entre 1 e N cuja raiz quadrada não tem parte decimal (ou seja, é exata). Antes de escrever o código pense sobre as propriedades matemáticas e algumas estratégias diferentes antes de decidir por uma.

- dica: qual o inverso de raiz quadrada? Como você pode tirar proveito disso?

## 16. mdc

Pesquise sobre o Algoritmo de Euclides para mdc entre dois números. Depois que entender como ele funciona implemente e teste seu programa.

<i>Entrada</i>	<i>Entrada</i>	<i>Entrada</i>	<i>Entrada</i>
1 4	4 10	8 12	20 25
<i>Saída</i>	<i>Saída</i>	<i>Saída</i>	<i>Saída</i>
1	2	4	5

## 17. triângulo de Floyd

O triângulo de Floyd é um triângulo que, dado um número N, é formado por N linhas, que são formadas por inteiros. A regra é que a 1ª linha é formada por 1 número, a 2ª formada por 2 e até a N-ésima formada por N números. Além disso, os números que compõe o triângulo são todos sucessores do anterior. Por exemplo, o triângulo de Floyd para N = 6 seria:

1

---

2 3

4 5 6

7 8 9 10

11 12 13 14 15

16 17 18 19 20 21

Faça um programa que receba um inteiro e imprima o triângulo de Floyd correspondente.

## Exercícios do URI

Problemas nº 1045, 1072, 1144, 1151

---