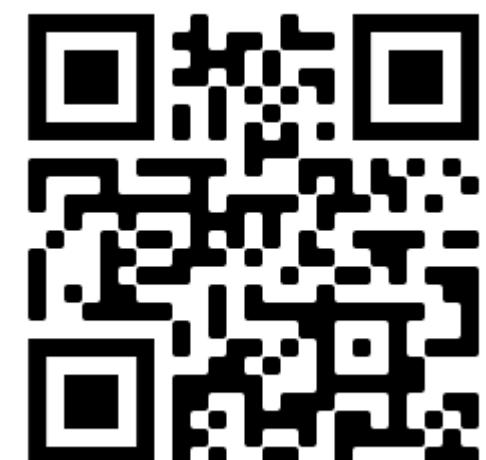


PROGRAMAÇÃO DE INTERNET

Prof. Richard Brosler

E-mail: richard.brosler@sp.senai.br

Grupo WhatsApp: <https://bit.ly/3K8jFIg>





AGENDA

- Revisão da aula passada
 - Operadores matemáticos
 - Operadores relacionais
 - Operadores lógicos
 - Operadores de bit-a-bit
 - Estruturas condicionais
 - SE condição ENTÃO
 - SE condição ENTÃO .. SENÃO
 - SEs aninhados
 - SELEÇÃO .. CASO
 - Exercícios
- Estruturas de repetição e Exercícios

OPERADORES MATEMÁTICOS

- Soma +
- Subtração -
- Multiplicação *
- Divisão / (cuidado que em algumas linguagens se dividirmos 2 inteiros, o resultado será um inteiro, ou seja, divisão inteira)
- Operador div Divisão inteira (visualG)
- Resto da divisão % (no caso do visualG é o mod)
- Exponenciação ** ou ^ (não são todas as linguagens que possuem um operador de exponenciação)

OPERADORES RELACIONAIS

- Maior que >
- Menor que <
- Igual a == (no caso do visualg é somente =)
- Diferente != (no caso do visualg é <>)
- Maior igual a >=
- Menor igual a <=

OPERADORES LÓGICOS

- Operador Lógico E: `&&` (visualG é o e)
- Tabela verdade para o operador Lógico `&&` (E ou AND)

A	B	Resultado ($A \wedge B$)
True	True	True
True	False	False
False	True	False
False	False	False

OPERADORES LÓGICOS

- Operador Lógico OU: || (visualG é o ou)
- Tabela verdade para o operador Lógico || (Ou ou OR)

A	B	Resultado ($A \vee B$)
True	True	True
True	False	True
False	True	True
False	False	False

OPERADORES LÓGICOS

- Operador Lógico Negação: ! (visualG é o não)
- Tabela verdade para o operador Lógico ! (Negação)

A	Resultado (\bar{A})
True	False
False	True

REPRESENTAÇÃO NUMÉRICA EM BINÁRIO

- Antes de conhecermos os operadores de bits, vamos conhecer um pouco mais sobre como o computador entende os números inteiros.
- Um número inteiro no computador internamente é visto no sistema binário, ou seja, temos que converter ele para base 2.
- $2 = 0010$
- $3 = 0011$
- $4 = 0100$
- $5 = 0101$

VAMOS CALCULAR

- Agora vamos calcular alguns números da base 10 para a base 2.
- Antes vamos entender a regra para conversão:
- Vamos converter o número 15 para binário:
- O resultado será 1111

Handwritten diagram illustrating the conversion of the decimal number 15 to binary using successive division by 2. The quotient 7 is circled in red. The remainders 1, 1, 1, 1 are circled in red and connected by a red arrow pointing downwards. The final remainder 0 is also circled in red.

VAMOS CALCULAR

- Se quisermos voltar da base 2 para a base 10, como temos que fazer?
- Vamos ao exemplo de 1110 para a base 10. Teremos o seguinte resultado:

1	1	1	0
*	*	*	*
2^3	2^2	2^1	2^0
8	4	2	0

- Somando-se $8 + 4 + 2 + 0 = 14$

OPERADORES BIT A BIT

- Operador AND : & (no visualG não temos)
- Vamos ao exemplo do operador & com 10 & 5. Primeiramente para entender, temos que converter os 2 números em binário para assim fazermos a conta:
- 10 em binário é 1010 e 5 em binário é 0101, logo, teremos a seguinte conta:

10	1	0	1	0
5	0	1	0	1
Resultado	0	0	0	0

- O resultado da expressão 10 & 5 será 0 (zero)

OPERADORES BIT A BIT

- Operador AND : & (no visualG não temos)
- Vamos outro exemplo, agora vamos fazer com 10 & 3
- 10 em binário é 1010 e 3 em binário é 11, logo, teremos a seguinte conta:

10	1	0	1	0
3	0	0	1	1
Resultado	0	0	1	0

- O resultado da expressão 10 & 3 será 2 (10 em binário)

OPERADORES BIT A BIT

- Operador OR : | (no visualG não temos)
- Vamos ao exemplo do operador | com 10 | 5 :
- 10 em binário é 1010 e 5 em binário é 0101, logo, teremos a seguinte conta:

10	1	0	1	0
5	0	1	0	1
Resultado	1	1	1	1

- O resultado da expressão 10 | 5 será 15 (15 em binário é 1111)

OPERADORES BIT A BIT

- Operador XOR (Ou exclusivo): \wedge (2 iguais = falso, 1 diferente = true) (no visualG não temos)
- Vamos ao exemplo do operador \wedge com $10 \wedge 7$:
- 10 em binário é 1010 e 7 em binário é 0111, logo, teremos a seguinte conta:

10	1	0	1	0
7	0	1	1	1
Resultado	1	1	0	1

- O resultado da expressão $10 \wedge 7$ será 13 (13 em binário é 1101)

OPERADORES BIT A BIT

- Operador Complemento : \sim (no visualG não temos)
- Vamos ao exemplo do operador \sim com ~ 10 :
- 10 em binário é 1010, teremos a seguinte conta:

10	1	0	1	0	
Resultado	0	1	0	1	

- O resultado da expressão ~ 10 será 5 (5 em binário é 0101)

OPERADORES BIT A BIT

- Operador Deslocamento de bit à esquerda: << (no visualG não temos)
- Vamos ao exemplo do operador << com 6 << 2 :
- 6 em binário é 0110, e vamos deslocar 2 bits à esquerda. logo, teremos a seguinte conta:

6	0	0	0	0	0	1	1	0
Resultado	0	0	0	1	1	0	0	0

- O resultado da expressão 6 << 2 será 24 (24 em binário é 0001 1000)

OPERADORES BIT A BIT

- Operador Deslocamento de bit à direita: `>>` (no visualG não temos)
- Vamos ao exemplo do operador `>>` com `12 >> 2` :
- 12 em binário é 1100, e vamos deslocar 2 bits à direita. logo, teremos a seguinte conta:

12	0	0	0	0	1	1	0	0	
Resultado	0	0	0	0	0	0	1	1	

- O resultado da expressão `12 >> 2` será 3 (3 em binário é 0011)

HIERARQUIA DOS OPERADORES

- Separadores
- Operadores unários
- Pré-incremento e pré-decremento
- Expoente
- Multiplicação / divisão
- Adição e subtração
- Deslocamento de bit
- Relacional exceto igualdade e diferente
- Igualdade e diferente
- Operador & (bit a bit)
- Operador ^ (bit a bit)
- Operador | (bit a bit)
- Operador lógico &&
- Operador Lógico ||
- Condicional ternário ? : (iremos ver)
- Atribuição = += -= *= /= %=
- Pós-incremento e pós-decremento

ESTRUTURAS CONDICIONAIS

SE CONDIÇÃO ENTÃO

- Quando desejamos condicionar o nosso programa para que em determinada situação seja executada, temos que utilizar uma estrutura condicional simples, composta ou aninhada. Nesse caso iremos ver ao simples.
- A estrutura do comando “Se” é a seguinte:
- SE condição ENTÃO
- Comandos1
- Comandos2
- ComandoN
- Fimse

ESTRUTURAS CONDICIONAIS

SE condição ENTÃO SENÃO

- A estrutura do comando “Se” é a seguinte:
- SE condição ENTÃO
- Comandos1
- Comandos2
- ComandoN
- SENÃO
- comando 4
- comando5
- Fimse

ESTRUTURAS CONDICIONAIS

SE condição ENTÃO SENÃO SE

- A estrutura do comando “Se” é a seguinte:
- SE condição ENTÃO
 - Comandos1
 - Comandos2
 - ComandoN
- SENÃO
 - SE condicao2 ENTAO
 - comando 4
 - comando 5
 - FIMSE
 - FIMSE

ESTRUTURAS CONDICIONAIS

SELEÇÃO .. CASO

- A estrutura do comando “seleção” é a seguinte:
- Escolha (variável)
- caso valor1
 - Comandos
- caso valor2
 - comandos
- outrocaso
 - comandos
- Fimescolha
 - Veja exemplo na página 51 do livro de lógica

ESTRUTURAS DE REPETIÇÃO

- Se você tivesse que montar um programa para mostrar a tabuada do 5, como você faria?
- Basicamente ficaria como essa imagem ao lado.
- Tente montar esse código no visualG com o que aprendeu até agora.
- O programa vai se chamar Aula04Tab01.
- Tempo para vocês montarem: 15 minutos

5	x	0	=	0
5	x	1	=	5
5	x	2	=	10
5	x	3	=	15
5	x	4	=	20
5	x	5	=	25
5	x	6	=	30
5	x	7	=	35
5	x	8	=	40
5	x	9	=	45
5	x	10	=	50

ESTRUTURAS DE REPETIÇÃO

- Se notarem, temos uma sequência de números que vai de zero à 10, e se multiplicarmos esse numero da sequencia, teremos o valor correspondente ao que está no igual....
- Correto?
- Agora utilizem uma variável inteira chamada intSeq que se inicie com zero, e a medida que imprime uma linha, aumenta 1 nessa intSeq.
- O resultado vai ser o mesmo não vai?
- Faça o código um novo algoritmo com o nome de Aula04Tab02, porém, nesse utilize o intSeq para mostrar a sequencia e para calcular o produto.

5	x	0	=	0
5	x	1	=	5
5	x	2	=	10
5	x	3	=	15
5	x	4	=	20
5	x	5	=	25
5	x	6	=	30
5	x	7	=	35
5	x	8	=	40
5	x	9	=	45
5	x	10	=	50

ESTRUTURAS DE REPETIÇÃO ENQUANTO (WHILE)

- A estrutura ENQUANTO é da seguinte forma:

```
enquanto (condicao) faça
    comandoa
    comando b
    ...
    comando n
fim enquanto
```

- Vamos fazer o exercício da tabuada agora com a estrutura de repetição ENQUANTO.
- Crie um novo algoritmo chamado de Aula04Tab03 e vamos à codificação...
- Digite o código que está na página 70 do livro de princípios de lógica de programação.

ESTRUTURAS DE REPETIÇÃO

REPITA ... ATÉ (REPEAT)

- Agora vamos para uma estrutura semelhante, porém o teste fica no final do bloco de repetição.
- Vamos ao exemplo, crie um novo algoritmo e chame de Aula04Tab04, agora redigite o código que está na página 71 do livro princípios de lógica de programação.
- Notou a diferença de ambos os códigos, o do enquanto e do repita? Se não notou, repare na condição de cada um deles.

ESTRUTURAS DE REPETIÇÃO PARA .. DE ..ATE ..FACA

- Agora vamos para outra estrutura semelhante, porém, essa estrutura implementa um contador com um terminador e opcionalmente o passo (de quanto em quanto).
- Vamos ao exemplo, crie um novo algoritmo e chame de Aula04Tab05, agora redigite o código que está na página 72 do livro princípios de lógica de programação.
- Notou a diferença dos códigos, o do enquanto, do repita e agora do passo? Se não notou, repare na condição de cada um deles.

EXERCÍCIOS

- Agora vamos exercitar para esquentar o dia. Rssss
- Todos os algoritmos abaixo deverão ser desenvolvidos utilizando as 3 estruturas de repetição. (enquanto, repita e para)
- 1) Montar um algoritmo para mostrar os números de 1 à 100.
- 2) Montar um algoritmo para mostrar os números de 100 à 1.
- 3) Montar um algoritmo para mostrar os 100 primeiros pares.
- 4) Montar um algoritmo para mostrar os múltiplos de 5 no intervalo de 1 a 500.
- 5) Montar um algoritmo para mostrar os números quadrados de 1 a 20. (Elevado a 2)

EXERCÍCIOS

- 6) Montar um algoritmo para listar os números de 1 a 100 e classifica-los em pares ou impares.
- 7) Montar um algoritmo que solicite 5 números inteiros, ao terminar, deverá mostrar qual foi o menor número digitado.
- 8) Montar um algoritmo que solicite 5 números inteiros, ao terminar, deverá mostrar qual foi o maior número digitado.
- 9) Montar um algoritmo que solicite 5 números inteiros, ao terminar, deverá mostrar a soma de todos os números digitados.
- 10) Montar um algoritmo que solicite 5 números inteiros, ao terminar, deverá mostrar a média calculada com base nos números digitados.

EXERCÍCIOS

- 11) montar um algoritmo que monte essa tela abaixo. Dica: Você deverá usar 2 laços de repetição. Um para a linha e dentro um para coluna.

1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8	1-9
2-1	2-2	2-3	2-4	2-5	2-6	2-7	2-8	2-9
3-1	3-2	3-3	3-4	3-5	3-6	3-7	3-8	3-9
4-1	4-2	4-3	4-4	4-5	4-6	4-7	4-8	4-9
5-1	5-2	5-3	5-4	5-5	5-6	5-7	5-8	5-9
6-1	6-2	6-3	6-4	6-5	6-6	6-7	6-8	6-9
7-1	7-2	7-3	7-4	7-5	7-6	7-7	7-8	7-9
8-1	8-2	8-3	8-4	8-5	8-6	8-7	8-8	8-9
9-1	9-2	9-3	9-4	9-5	9-6	9-7	9-8	9-9

EXERCÍCIOS

- 12) montar um algoritmo que monte essa tela abaixo. Dica: Você deverá usar 2 laços de repetição. Um para a linha e dentro um para coluna.

1 - 1
2 - 1 2 - 2
3 - 1 3 - 2 3 - 3
4 - 1 4 - 2 4 - 3 4 - 4
5 - 1 5 - 2 5 - 3 5 - 4 5 - 5
6 - 1 6 - 2 6 - 3 6 - 4 6 - 5 6 - 6
7 - 1 7 - 2 7 - 3 7 - 4 7 - 5 7 - 6 7 - 7
8 - 1 8 - 2 8 - 3 8 - 4 8 - 5 8 - 6 8 - 7 8 - 8
9 - 1 9 - 2 9 - 3 9 - 4 9 - 5 9 - 6 9 - 7 9 - 8 9 - 9

EXERCÍCIOS

- 13) montar um algoritmo que monte essa tela abaixo. Dica: Você deverá usar 2 laços de repetição. Um para a linha e dentro um para coluna. Para ficar na posição correta, substitua a posição por 3 brancos.

1-1	1-2	1-3	1-4	1-5	1-6	1-7	1-8	1-9
2-2	2-3	2-4	2-5	2-6	2-7	2-8	2-9	
	3-3	3-4	3-5	3-6	3-7	3-8	3-9	
	4-4	4-5	4-6	4-7	4-8	4-9		
		5-5	5-6	5-7	5-8	5-9		
			6-6	6-7	6-8	6-9		
				7-7	7-8	7-9		
					8-8	8-9		
						9-9		