

Exercícios 04 :: Comandos de Repetição 2

Instruções Gerais

- Os exercícios são de resolução individual.
- Cria uma pasta para a lista e faça cada exercício em um arquivo distinto.
- Utilize a extensão .c e o compilador gcc. Utilize o editor VS Code ou outro de sua preferência.
- Não é permitido o uso de recursos que ainda não foram abordados na disciplina até o momento da publicação desta lista. Esta lista considera:**
 - Comandos de seleção e repetição.**

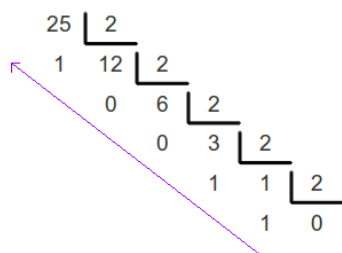
- Escreva um programa que, dado um número inteiro (positivo ou negativo), troca o último dígito pelo primeiro. OBS: não basta imprimir o número dessa forma, é preciso inverter o número em uma variável **int**.

```
Informe o número: -2567
> -7562
```

- Escreva um programa que imprime a tabela ASC com valores em decimal (%03d), hexadecimal (%03X) e o caractere (%c). Imprima apenas os caracteres 33 ao 125, separados em 3 colunas (cada uma conterá 31 elementos), similar ao seguinte exemplo:

DEC	HEX	CHR	DEC	HEX	CHR	DEC	HEX	CHR
033	021	!	064	040	@	095	061	—
034	022	"	065	041	A	096	062	`
035	023	#	066	042	B	097	063	a
036	024	\$	067	043	C	098	064	b
...				
061	03D	=	092	05C	\	123	07D	{
062	03E	>	093	05D]	124	07E	
063	03F	?	094	05E	^	125	07F	}

- Escreva um programa que, dado um número inteiro positivo, o imprime em notação **binária**. O processo de obtenção é similar à extrair os dígitos de um número: **divisões sucessivas por 2**, até que o **quociente seja 0** (zero). A cada divisão, o **resto** indica um **dígito binário**. Entretanto, assim como no algoritmo de obter os dígitos, os dígitos binários estarão em **ordem inversa**. Logo, será preciso remontar o número binário (na ordem correta) em uma variável inteira. Observe o exemplo:



25_(decimal) = 11001_(binário)

4. Escreva um programa que, dado um número inteiro positivo, verifica se o mesmo é um Número Perfeito. Um número perfeito é igual à soma de seus divisores positivos. Exemplo: $6 = 1 + 2 + 3$,

```
Informe o numero: 6
Divisores de 6: 1 + 2 + 3 = 6
Numero perfeito? Sim
```

5. Escreva um programa que imprime uma caixa de caracteres '[' com as dimensões informadas. O ler as dimensões da caixa (largura e altura). Observe o exemplo.

```
Altura x Largura: 4 5
```

```
[[] [] [] [] []]
[] [] [] [] []
[] [] [] [] []
[] [] [] [] []
```

6. Escreva um programa que imprime um triângulo de caracteres '[' com o número de linhas informado. Observe o exemplo.

```
Linhas: 5
```

```
[]
[] []
[] [] []
[] [] [] []
[] [] [] [] []
```

7. Escreva um programa que imprime um triângulo invertido de caracteres '[' , alinhado à direita, com o número de linhas informado. Observe o exemplo.

```
Linhas: 5
```

```
      []
     [] []
    [] [] []
   [] [] [] []
  [] [] [] [] []
```

8. Escreva um programa que imprime um triângulo centralizado de caracteres '[' . Observe o exemplo.

```
Linhas: 6
```

```
      []
     [] []
    [] [] []
   [] [] [] []
  [] [] [] [] []
 [] [] [] [] [] []
```