

Lista de Exercícios no. 03A :: Laços++

Instruções Gerais

- Os exercícios são de resolução individual.
- Sempre tente resolver os exercícios por conta própria, mesmo aqueles que já tenham sido feitos pelo professor em aula.
- Crie uma pasta para a lista e faça cada exercício em um arquivo distinto.
- Utilize a extensão .py e o editor VS Code (ou outro de sua preferência).
- **Não é permitido o uso de recursos da linguagem ou bibliotecas que ainda não foram abordados na disciplina até o momento da publicação desta lista.**

1. 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
```

2. 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	}

3. Escreva um programa que, dado um número inteiro (positivo ou negativo) entre -99 e 99, o imprime por extenso. Caso o número esteja fora desse intervalo, o programa deve informar um erro. Dica: como o tamanho do número é conhecido (2 dígitos), você pode previamente desmontá-lo, colocando cada dígito em variáveis separadas.

```
Informe o número: -57
> cinquenta e sete (negativo)
```

4. Escreva um programa que, dado um número inteiro (positivo ou negativo) entre -999 e 999, o imprime por extenso. Caso o número esteja fora desse intervalo, o programa deve informar um erro.

```
Informe o número: 572
> quinhentos e setenta e dois (positivo)
```

5. 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
```

6. Escreva um programa que, dado um número inteiro positivo, imprime seus fatores primos.

```
Informe o numero:
> 132
2 x fator 2
1 x fator 3
1 x fator 11
```

7. Escreva um programa que, dado dois números inteiros positivos (A e B), imprime o MDC (Máximo Divisor Comum) entre ambos. O MDC é o maior número que divide A e B (divisão exata, de resto zero). Dica: o máximo divisor comum entre A e B está, necessariamente, entre 2 e o menor entre A e B, isto é, $\text{MENOR}(A, B)$. Se você não encontrar um divisor comum entre A e B, então $\text{MDC} = 1$.

```
Informe os numeros: 30 12
> Maximo divisor comum: 6
```

Observe o processo

```
-----
Divisores de 30: 1, 2, 3, 5, 6, 10, 15, 30
Divisores de 12: 1, 2, 3, 4, 6, 12
                        ^ MDC
```

8. Escreva um programa que, dado dois números inteiros positivos (A e B), imprime o MMC (Mínimo Múltiplo Comum) entre ambos. O MMC é o menor número que resulta da multiplicação de A e B por um número (não necessariamente igual para ambos). Dica: o mínimo múltiplo comum entre A e B pode ser encontrado verificando se algum múltiplo do $\text{MAIOR}(A, B)$ é divisível pelo $\text{MENOR}(A, B)$. Sempre há um MMC entre dois números quaisquer.

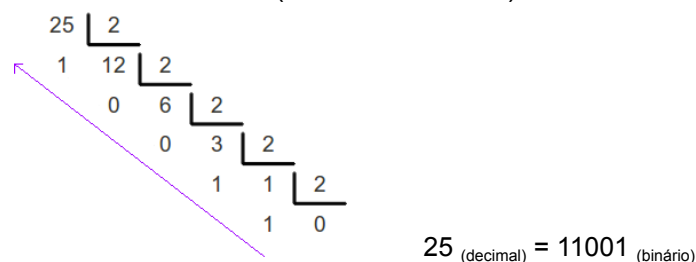
```
Informe os numeros: 30 12
> Minimo multiplo comum: 60
```

Observe o processo

```
-----
Múltiplos de 30: 30, 60, 90, 120, 150, ...
Múltiplos de 12: 12, 24, 36, 48, 60, ...
                  x1  x2  x3  x4  x5  ...
```

O MMC entre 30 e 12 é 60.

9. 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:



10. Escreva um programa que imprime uma caixa de caracteres com as dimensões informadas. O programa deve ler o caractere, bem como, as dimensões da caixa (largura e altura). **Você só pode usar 1 laço (não deve usar laços aninhados, isto é, laço dentro de laço). Também não pode utilizar operadores e funções de strings.**

Exemplo:

Caractere: X

Largura: 5

Altura: 3

```
XXXXX
XXXXX
XXXXX
```

11. Escreva um programa que imprime um triângulo de caracteres com o número de linhas informado. **Você só pode usar 1 laço (não deve usar laços aninhados, isto é, laço dentro de laço). Também não pode utilizar operadores e funções de strings.**

Informe a quantidade de linhas: 5

```
X
XX
XXX
XXXX
XXXXX
```

12. Escreva um programa que imprime um triângulo de caracteres invertido com o número de linhas informado. **Você só pode usar 1 laço (não deve usar laços aninhados, isto é, laço dentro de laço). Também não pode utilizar operadores e funções de strings.**

Informe a quantidade de linhas: 5

```
XXXXX
XXXX
XXX
XX
X
```

13. Escreva um programa que imprime um triângulo de caracteres invertido, alinhado à direita, com o número de linhas informado. **Você só pode usar 1 laço (não deve usar laços aninhados, isto é, laço dentro de laço). Também não pode utilizar operadores e funções de strings.**

Informe a quantidade de linhas: 5

```

    X
   XX
  XXX
 XXXX
XXXXX
```

14. Escreva um programa que imprime um triângulo de caracteres centralizado. **Você só pode usar 1 laço (não deve usar laços aninhados, isto é, laço dentro de laço). Também não pode utilizar operadores e funções de strings.**

Informe a quantidade de linhas: 5

```

    X
   XXX
  XXXXX
 XXXXXXX
XXXXXXXXX
```