

### Lista de exercícios - 16

#### Exercícios: laços de repetição com while e for

Para cada um dos exercícios, crie um arquivo fonte Python com o respectivo nome de acordo com as regras já determinadas nas listas de exercícios anteriores.

**Observação:** todos os exercícios devem ser resolvidos com laços de repetição utilizando o comando **while** ou o comando **for**. Em cada exercício, você pode decidir qual dos dois comandos é mais apropriado.

#### Questões:

1. **Tabela de conversão de temperaturas.** Escreva um programa Python que mostre uma tabela de conversão de temperaturas em graus Celsius e graus Fahrenheit. A tabela deve incluir em suas linhas todas as temperaturas entre 0 e 100 graus Celsius que sejam múltiplas de 10 graus Celsius. Inclua os cabeçalhos apropriados e tabulações para suas colunas. Pesquise na internet sobre a fórmula de conversão de temperaturas Celsius para Fahrenheit. Pesquise a matéria de Física para saber sobre escalas de temperatura Celsius e Fahrenheit.

**Dica:** dentro das strings na função print você pode usar o caracter especial “\t”, que insere um espaço de tabulação na string (equivalente a usar a tecla “tab” quando digitamos um texto).

2. **Valor das entradas.** Um determinado zoológico estipula o valor da entrada baseado na idade do visitante. Visitantes com até dois anos de idade não precisam pagar. Crianças entre 3 e 12 anos de idade pagam R\$ 14.00. Idosos com 65 anos ou mais pagam R\$ 18.00. Todos os demais pagam R\$ 23.00. Crie um programa que inicia lendo as idades, uma por uma, de um grupo de pessoas. O usuário deve entrar uma linha em branco para indicar que não há mais pessoas no grupo. Depois disso, seu programa deve exibir uma mensagem informando o preço total de todas as entradas para o grupo. O valor deve ser exibido com duas casas decimais.
3. **Tabela de multiplicação.** Neste exercício você deve criar uma tabela de multiplicação mostrando os produtos de todos os inteiros de 1 vezes 1 até 10 vezes 10. Sua tabela deve incluir uma linha de cabeçalho com números de 1 a 10, e também uma coluna com os mesmos números. A saída esperada do programa deve ser semelhante ao mostrado abaixo:

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

Ao desenvolver seu programa, talvez você ache conveniente exibir um valor com print sem mover para a próxima linha. Isso pode ser feito adicionando-se **end=""** como último parâmetro da função print (Exemplo: `print(x, end="")`).

4. **Fatoração numérica.** A fatoração de um número inteiro  $n$  pode ser feita por meio de números primos de acordo com o procedimento descrito abaixo:

Inicialize fator com valor 2

Enquanto fator for menor ou igual a  $n$ , faça

    Se  $n$  for divisível por fator então

        Concluimos que fator faz parte da fatoração de  $n$

        Faça divisão inteira de  $n$  por fator

    Senão

        Incremente fator em uma unidade

Escreva um programa Python que lê um número inteiro do usuário. Se o valor fornecido pelo usuário é menor do que 2, seu programa deve exibir uma mensagem de erro. Caso contrário, seu programa deve exibir os números primos que podem ser multiplicados para formar o número  $n$ , com um fator exibido em cada linha. Por exemplo:

Digite um número inteiro (maior ou igual a 2): 72

2  
2  
2  
3  
3

5. **Binário para decimal.** Escreva um programa Python que converte um número binário (base 2) para decimal (base 10). Seu programa deve iniciar lendo um número binário como uma string. Então, ele deve computar o número decimal equivalente processando cada dígito do número binário. Finalmente seu programa deve exibir uma mensagem informando o número decimal calculado.

**Dica:** para acessar cada um dos dígitos da string, você pode utilizar o laço for da seguinte maneira: "**for letra in palavra:**". Neste caso, a variável palavra é uma string e a variável letra é cada um dos caracteres da string.

6. **Decimal para binário.** Escreva um programa Python que converte um número decimal (base 10) para o correspondente número binário (base 2). Leia o número decimal como um número inteiro fornecido pelo usuário. Depois disso, use o algoritmo de divisão mostrado abaixo para fazer a conversão. Quando o algoritmo terminar, a variável *result* contém a representação binária do número. Ao final exiba uma mensagem informando o valor de *result*.

Inicialize *result* como uma string vazia

Seja  $q$  o número decimal a ser convertido

**Repita**

$r$  recebe o resto da divisão de  $q$  por 2

    converta  $r$  para uma string e adicione no início de *result*

    faça a divisão inteira de  $q$  por 2 (descartando o resto) e guarde o resultado em  $q$

**Até que  $q$  seja igual a zero**