



1. Escreva uma função que recebe um parâmetro inteiro PN passado por valor e retorna 0 (zero) se o número for primo, caso contrário retorna algum valor diferente de zero. O cabeçalho desta função é:

```
def EPrimo (PN) :
```

Para testar a função acima escreva um programa que leia um número inteiro Max obrigatoriamente maior que 0 e, em seguida, preencha um vetor (lista) com todos os números primos menores que o valor Max.

2. Escreva uma função que calcule e retorne a média dos valores contidos em uma lista passada como parâmetro e que deverá estar preenchida com valores numéricos inteiros ou reais. Note bem, a função deve receber a lista previamente preenchida e não preencher a lista. Escreva um programa para testar essa função.

```
def CalculaMedia (Lista) :
```

3. Escreva uma função que leia do teclado e retorne um número inteiro. Essa função deve receber dois parâmetros inteiros passados por valor (prMin e prMax). Esta função deve ser declarada com o seguinte cabeçalho:

```
def LeNumInt (prMin, prMax) :
```

e seguir as seguintes regras:

- se prMin != 0 e prMax != 0 o número retornado deve estar no intervalo fechado: prMin <= Retorno <= prMax;
- se prMin != 0 e prMax == 0 o número retornado deve ser maior ou igual a prMin: Retorno >= prMin;
- se prMin == 0 e prMax != 0 o número retornado deve ser menor ou igual a prMax: Retorno <= prMax;

4. Escreva o código de uma função que seja capaz ler N valores inteiros e armazená-los em um vetor (lista). Esta função deve ser declarada com o seguinte cabeçalho:

```
def LeVetorInt (N) :
```

N é o parâmetro que recebe o tamanho do vetor. O vetor gerado deve ser retornado na forma de lista (não tupla)

Escreva também um programa para usar e testar esta função criada.

5. Escreva o código de uma função que seja capaz ler N valores reais e armazená-los em um vetor (lista). Esta função deve ser declarada com o seguinte cabeçalho:

```
def LeVetorReal (N) :
```

N é o parâmetro que recebe o tamanho do vetor. O vetor gerado deve ser retornado na forma de lista (não tupla)

Escreva também um programa para usar e testar esta função criada.

6. Escreva uma função que gere um vetor (lista) de tamanho prN com valores inteiros aleatórios entre prMin e prMax, passados para a função. Antes de realizar a geração dos valores verificar se prMin é menor ou igual a prMax. Se isto não ocorrer, a função deve trocar seus valores antes de gerar os aleatórios.

```
def GeraVetor (prN, prMin, prMax) :
```

Escreva também um programa para usar e testar esta função criada.

7. Considere que o dado "3177-4" é o código de um produto. O último dígito é chamado de "dígito verificador". Escreva uma função que recebe como parâmetro um número inteiro entre 1000 e 9999 (4 dígitos) que é um código de produto e seu dígito verificador calculado pelo método chamado de módulo 7 conforme mostrado abaixo:

Exemplo de cálculo do DV do Código do Produto para o código 3177

Dígitos Os dígitos do número recebido pela função devem ser desmembrados através de divisões sucessivas por potências de 10 (1000, 100 e 10)	3	1	7	7
Pesos Cada dígito terá um peso distinto começando em 2, como mostrado aqui ao lado.	5	4	3	2
Cada dígito deve ser multiplicado pelo peso	15	4	21	14
O próximo passo é somar todos os valores acima	15 + 4 + 21 + 14 = 54			
Calcule o resto da somatória por 7	Resto de 54 por 7 = 5			

Portanto o DV do código 3177 é 5 e o código completo ficará sendo 3177-5

Para as questões 6 e 7 a seguir consulte o link <http://dan-scientia.blogspot.com.br/2013/03/os-digitos-verificadores-do-cpf-cnpj.html> para saber como são calculados os dígitos verificadores do CPF e do CNPJ brasileiros.

8. Escreva uma função que receba um número de CPF (sem pontos e traços) e calcule e retorne seus dígitos verificadores (são dois calculados separadamente e exibidos como se fossem um valor de dois dígitos).
9. Escreva uma função que receba um número de CNPJ (sem pontos e traços) e calcule e retorne seus dígitos verificadores (são dois calculados separadamente e exibidos como se fossem um valor de dois dígitos).
10. Considere a função com o cabeçalho abaixo. Essa função deve gerar e retornar uma senha constituída exclusivamente por números inteiros num total de dígitos dado pelo parâmetro "tamanho". Escreva um programa para testar essa função.

```
def GeraSenhaNumerica(tamanho):
```

11. Considere a função com o cabeçalho abaixo. Essa função deve gerar e retornar uma senha constituída exclusivamente por letras com o total de caracteres dado pelo parâmetro "tamanho".

```
def GeraSenhaAlfa(tamanho, caps):
```

Como pode ser visto acima, esta função terá um segundo parâmetro chamado "caps" que indicará se as letras devem ser maiúsculas, minúsculas ou ambas. Esse parâmetro caps deve ser usado da seguinte forma. Escreva um programa para testar essa função.

- se caps == 1 todas as letras devem ser maiúsculas
- se caps == 2 todas as letras devem ser minúsculas
- se caps for qualquer outro valor as letras geradas na senha devem ser aleatoriamente misturadas maiúsculas e minúsculas.

12. Considere a função com o cabeçalho abaixo. Essa função deve gerar e retornar uma senha alfanumérica, ou seja, constituída por letras e números com o total de caracteres dado pelo parâmetro "tamanho".

```
def GeraSenhaAlfaNum(tamanho, caps):
```

Como pode ser visto acima, esta função terá um segundo parâmetro chamado "caps" que indicará se as letras devem ser maiúsculas, minúsculas ou ambas. Esse parâmetro caps deve ser usado da seguinte forma. Escreva um programa para testar essa função.

- se caps == 1 todas as letras devem ser maiúsculas
- se caps == 2 todas as letras devem ser minúsculas
- se caps for qualquer outro valor as letras geradas na senha devem ser aleatoriamente misturadas maiúsculas e minúsculas.