

Computação I - Python

Aula 11 - Teórica: Entrada e Saída e Programa Principal

João C. P. da Silva

Carla A. D. M. Delgado

Ana Luisa Duboc

Dept. Ciência da Computação - UFRJ

Entrada e Saída de dados - Interação com o Usuário

Saída de dados - escrevendo

- **print**: Usado para escrever informações na saída padrão.
- De modo geral a saída padrão é o console (interpretador)

```
1 >>> print("Qual o seu nome?")
2 Qual o seu nome?
```

```
1 def soma(numero1 , numero2):
2
3     """ Funcao que soma dois numeros inteiros
4     Parametro de entrada: int,int
5     Valor de retorno: int"""
6
7     return numero1 + numero2
```

```
1 >>> print("A soma de 2 e 3 e " ,soma(2,3))
2 A soma de 2 e 3 e 5
```

Entrada e Saída de dados - Interação com o Usuário

Saída de dados - escrevendo

- Usado o **print** para escrever informações na tela.

```
1 >>> print("Qual o seu nome?")
2 Qual o seu nome?
```

```
1 def soma(numero1 , numero2):
2
3     """ Funcao que soma dois numeros inteiros
4     Parametro de entrada: int ,int
5     Valor de retorno: int """
6
7     return print(numero1 + numero2)
```

ATENÇÃO: NÃO USE O PRINT DESTA FORMA !

- Comandos para leitura e escrita de dados devem ser usados de forma a **não interferir com o funcionamento das funções** que não tem o propósito específico de realizar a entrada ou saída de dados.
- Lembre que uma função pode ser usada por diversas outras!

Entrada e Saída de dados - Interação com o Usuário

Escrevendo informações com o print

- `print(exp1,exp2,exp3)` → próximo print na linha seguinte.
- `print(exp1,exp2,exp3,end="")` → próximo print na mesma linha.

```
1 print("Eu")
2 print("saio", "no bloco")
3 print("Suvaco do Cristo")
```

Eu
saio no bloco
Suvaco do Cristo

```
1 print("Eu")
2 print("saio", "no bloco",end="")
3 print("Suvaco do Cristo")
```

Eu
saio no bloco Suvaco do Cristo

Entrada e Saída de dados - Interação com o Usuário

Escrevendo informações com o print

- `"\n"` é usado para quebra de linha quando usado dentro de uma string.

```
1 >>> print("Meu nome e Jose \n e eu tenho 10 anos")
2 Meu nome e Jose
3 e eu tenho 10 anos
4
5 >>> print("A soma de 2 e 3 e : \n", soma(2,3))
6 A soma de 2 e 3 e :
7 5
8
9 >>> print("A soma de 2 e 3 e : ", soma(2,3))
10 A soma de 2 e 3 e : 5
```

Entrada e Saída de dados - Interação com o Usuário

Escrevendo informações com o print

- `str.format` é uma função muito usada com o `print`, para formatar a string que será impressa na tela.

```
1 >>> print(str.format("A soma de 2 e 3 e : {0:5d}", soma(2,3)))
2 A soma de 2 e 3 e :      5
3
4 >>> print(str.format("A soma de {0:3.2f} e {1:3.2f} e : {2:5.2f}",
5     f1, f2, soma(f1, f2)))
A soma de  2.50 e 30.00 e :    32.50
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que peça ao usuário para digitar 2 números inteiros e imprima na tela o resultado da função soma para estes dois números.

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que peça ao usuário para digitar 2 números inteiros e imprima na tela o resultado da função soma para estes dois números.

```
1 def soma(numero1 , numero2):  
2  
3     """ Funcao que soma dois numeros inteiros  
4     Parametro de entrada: int ,int  
5     Valor de retorno: int """  
6  
7     return numero1 + numero2
```


Entrada e Saída de dados - Interação com o Usuário

Faça um programa que peça ao usuário para digitar 2 números inteiros e imprima na tela o resultado da função soma para estes dois números.

```
1 def soma(numero1 , numero2):  
2  
3     """ Funcao que soma dois numeros inteiros  
4     Parametro de entrada: int,int  
5     Valor de retorno: int """  
6  
7     return numero1 + numero2
```

- vamos definir a função principal do nosso programa.

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que peça ao usuário para digitar 2 números inteiros e imprima na tela o resultado da função soma para estes dois números.

```
1 def soma(numero1 , numero2):  
2  
3     """ Funcao que soma dois numeros inteiros  
4     Parametro de entrada: int,int  
5     Valor de retorno: int """  
6  
7     return numero1 + numero2
```

- vamos definir a função principal do nosso programa.

```
1 def main():  
2  
3     """ Funcao Principal """  
4  
5     print(str.format("A soma de 2 e 3 e:",soma(2,3)))  
6  
7     if __name__ == "__main__":  
8         main()
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que peça ao usuário para digitar 2 números inteiros e imprima na tela o resultado da função soma para estes dois números.

```
1 def soma(numero1 , numero2):  
2  
3     """ Funcao que soma dois numeros inteiros  
4     Parametro de entrada: int ,int  
5     Valor de retorno: int """  
6  
7     return numero1 + numero2
```

- vamos definir a função principal do nosso programa.

```
1 def main():  
2  
3     """ Funcao Principal """  
4  
5     print(str.format("A soma de 2 e 3 e:", soma(2,3)))  
6  
7     if __name__ == "__main__":  
8         main()
```

Observe que ao executar o programa, o resultado aparece direto na tela, sem precisar chamar a função *main()*.

Entrada e Saída de dados - Interação com o Usuário

Por que função `main()` ?

- Considere que você salvou as funções `soma` e `main` no arquivo `MeuPrograma.py`

```
1 def soma(numero1 , numero2):
2     """ Funcao que soma dois numeros inteiros
3     Parametro de entrada: int,int
4     Valor de retorno: int"""
5     return numero1 + numero2
6
7 def main():
8     """ Funcao Principal"""
9
10    print(str.format("A soma de 2 e 3 e: {0:3d}",soma(2,3)))
11
12 if __name__ == "__main__":
13    main()
```

Entrada e Saída de dados - Interação com o Usuário

Por que função `main()` ?

- Considere que você salvou as funções `soma` e `main` no arquivo `MeuPrograma.py`

```
1 def soma(numero1 , numero2):
2     """ Funcao que soma dois numeros inteiros
3     Parametro de entrada: int,int
4     Valor de retorno: int"""
5     return numero1 + numero2
6
7 def main():
8     """ Funcao Principal"""
9
10    print(str.format("A soma de 2 e 3 e: {0:3d}",soma(2,3)))
11
12 if __name__ == "__main__":
13    main()
```

- Suponha que você fez a função `vezes5`, salvando-a no arquivo `Auxiliar.py`:

```
1 from MeuPrograma import *
2
3 def vezes5(numero1 , numero2):
4
5     """ Multiplica por 5 o resultado da funcao soma
6     Parametros de entrada: int,int
7     Valor de retorno: int"""
8
9     return 5*soma(numero1 , numero2)
```

Entrada e Saída de dados - Interação com o Usuário

Por que função `main()` ?

- Note que ao invés de copiar a função `soma` do arquivo `MeuPrograma.py` para `Auxiliar.py`, apenas importamos as funções de `MeuPrograma.py` para `Auxiliar.py`

```
1 def soma(numero1 , numero2):
2     """ Funcao que soma dois numeros inteiros
3     Parametro de entrada: int ,int
4     Valor de retorno: int"""
5     return numero1 + numero2
6
7 def main():
8     """ Funcao Principal"""
9
10    print(str.format("A soma de 2 e 3 e: {0:3d}",soma(2,3)))
11
12 if __name__ == "__main__":
13    main()
```

```
1 from MeuPrograma import *
2
3 def vezes5(numero1 , numero2):
4
5     """ Multiplica por 5 o resultado da funcao soma
6     Parametros de entrada: int ,int
7     Valor de retorno: int"""
8
9     return 5*soma(numero1 , numero2)
```

Entrada e Saída de dados - Interação com o Usuário

Por que função `main()` ?

- Se você executar `MeuPrograma.py`, aparecerá "A soma de 2 e 3 é: 5".
- Se você executar `Auxiliar.py`, nada acontecerá, a menos que você chame a função `vezes5`. Por exemplo, `vezes5(2,3)` retornará 25.
- Por que isto ocorre ?

```
1 def soma(numero1 , numero2):
2     """ Funcao que soma dois numeros inteiros
3     Parametro de entrada: int,int
4     Valor de retorno: int"""
5     return numero1 + numero2
6
7 def main():
8     """ Funcao Principal"""
9
10    print(str.format("A soma de 2 e 3 e: {0:3d}",soma(2,3)))
11
12 if __name__ == "__main__":
13    main()
```

```
1 from MeuPrograma import *
2
3 def vezes5(numero1 , numero2):
4
5     """ Multiplica por 5 o resultado da funcao soma
6     Parametros de entrada: int,int
7     Valor de retorno: int"""
8
9     return 5*soma(numero1 , numero2)
```

Entrada e Saída de dados - Interação com o Usuário

Por que função `main()` ?

- Ao definir a função `main()` em **MeuPrograma.py**, estamos requerendo que os comandos dentro dela sejam executados ao executarmos este arquivo.
- O comando `if __name__ == "__main__":` serve para verificar se estamos ou não rodando **MeuPrograma.py** diretamente.
- Se estivermos, o teste do `if` será `True` e a função `main()` será executada.
- É isso que ocorre quando executamos **MeuPrograma.py**.

```
1 def soma(numero1 , numero2):
2     """Funcao que soma dois numeros inteiros
3     Parametro de entrada: int,int
4     Valor de retorno: int"""
5     return numero1 + numero2
6
7 def main():
8     """Funcao Principal"""
9
10    print(str.format("A soma de 2 e 3 e: {0:3d}",soma(2,3)))
11
12 if __name__ == "__main__":
13    main()
```


Entrada e Saída de dados - Interação com o Usuário

Por que função `main()` ?

- Já quando estamos executando **Auxiliar.py**, importamos as funções do arquivo **MeuPrograma.py**.
- Mas o comando "`if __name__ == '__main__':`" não será satisfeito (será **False**), uma vez que quem está sendo executado é **Auxiliar.py** e não **MeuPrograma.py**.
- Logo a função `main()` não será executada, e a resposta de **Auxiliar.py** será nada a menos que se chame a função `vezes5`, ou alguma função do arquivo importado.

```
1 from MeuPrograma import *
2
3 def vezes5(numero1 , numero2):
4
5     """ Multiplica por 5 o resultado da funcao soma
6     Parametros de entrada: int ,int
7     Valor de retorno: int """
8
9     return 5*soma(numero1 , numero2)
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que peça ao usuário para digitar 2 números inteiros e imprima na tela o resultado da função soma para estes dois números.

```
1 def soma(numero1, numero2):
2     """ Funcao que soma dois numeros inteiros
3     Parametro de entrada: int, int
4     Valor de retorno: int """
5     return numero1 + numero2
6
7 def main():
8     """ Funcao Principal """
9
10    print(str.format("A soma de 2 e 3 e: {0:3d}", soma(2,3)))
11
12 if __name__ == "__main__":
13    main()
```

Note que chamamos a função *soma* com valores fixos (2 e 3).
Como fazer para que o usuário possa entrar com dois valores quaisquer?

Entrada e Saída de dados - Interação com o Usuário

Entrada de dados - input

- **input(texto)** : permite que você entre com uma string

```
1 >>> numero1= input("Qual o primeiro numero? ")
2 Qual o primeiro numero? 7
3
4 >>> numero1
5 '7'
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que peça ao usuário para digitar 2 números inteiros e imprima na tela o resultado da função soma para estes dois números.

```
1 def soma(numero1 , numero2):
2     """ Funcao que soma dois numeros inteiros
3     Parametro de entrada: int,int
4     Valor de retorno: int"""
5     return numero1 + numero2
6
7 def main():
8     """ Funcao Principal"""
9     numero1 = int(input( "Qual o primeiro numero? " ))
10    numero2 = int(input( "Qual o segundo numero? " ))
11    print(str.format("A soma de {d} e {d} eh: ", numero1 , numero2 , soma(numero1 , numero2)))
12 if __name__ == "__main__":
13    main()
```

Execute o programa !

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que peça ao usuário para digitar uma quantidade indeterminada de números até que seja digitado um número negativo. O programa deve armazenar os números lidos numa lista. Depois disso, o programa deve imprimir a lista e a soma de todos os números da lista.

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que peça ao usuário para digitar uma quantidade indeterminada de números até que seja digitado um número negativo. O programa deve armazenar os números lidos numa lista. Depois disso, o programa deve imprimir a lista e a soma de todos os números da lista.

```
1 def main():
2
3     """Programa que le uma quantidade indeterminada de numeros e os armazena em uma lista
4     . Imprime a soma de todos os numeros da lista. Funcao Principal"""
5
6     lista = [ ]
7     numero = int(input( "Digite um numero: "))
8     while numero >= 0:
9         lista.append(lista , numero)
10        numero = int(input( "Digite um numero: "))
11    print(lista)
12    print(str.format("A soma dos numeros e {:d}",sum(lista)))
13 if __name__ == "__main__":
14     main()
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que peça ao usuário para digitar uma quantidade indeterminada de números até que seja digitado um número negativo. O programa deve armazenar os números lidos numa lista. Depois disso, o programa deve imprimir a lista e a soma de todos os números da lista.

```
>>>
Digite um número:1
Digite um número:4
Digite um número:7
Digite um número:8
Digite um número:9
Digite um número:-1
[1, 4, 7, 8, 9]
A soma dos números é: 29
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 1º Passo: Como ler usando o `input()` ?

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 1º Passo: Como ler usando o `input()` ?

Lendo no formato de lista:

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 1º Passo: Como ler usando o input() ?

Lendo no formato de lista:

```
1 def main():
2
3     """ Primeira Opcao de Programa – Funcao Principal """
4
5     matriz = input(" Digite uma matriz : ")
6 if __name__ == "__main__":
7     main()
```

>>>

Digite uma matriz: [[1,2],[3,4]]

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 1º Passo: Como ler usando o `input()` ?

Lendo elemento por elemento:

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 1º Passo: Como ler usando o `input()` ?

Lendo elemento por elemento:

```
1 def leMat(numLinhas , numColunas ) :
2     """ Le uma matriz elemento a elemento
3     Parametros de entrada: int (numero de linhas), int (numero de colunas)
4     Valor de retorno: list """
5     matriz = [] #cria uma lista vazia para guardar a matriz
6     for i in range(numLinhas):
7         linha = [ ] #cria uma nova linha
8         for j in range(numColunas):
9             valor = int(input(" Digite o elemento ["+str(i)+"] ["+ str(j)+"] : "))
10            list.append(linha , valor) #adiciona um elemento na linha
11            list.append(matriz , linha) #adiciona linha pronta na matriz
12        return matriz
13
14
15 def main():
16     """ Exercício Conta pares – programa principal """
17     numLinhas = int(input(" Digite o numero de linhas : "))
18     numColunas = int(input(" Digite o numero de colunas : "))
19     matriz = leMat(numLinhas , numColunas)
20     if __name__ == "__main__":
21         main()
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 1º Passo: Como ler usando o `input()` ?

Lendo elemento por elemento:

```
1 >>>
2 Digite o numero de linhas : 2
3 Digite o numero de colunas : 3
4 Digite o elemento [0][0]: 1
5 Digite o elemento [0][1]: 2
6 Digite o elemento [0][2]: 3
7 Digite o elemento [1][0]: 4
8 Digite o elemento [1][1]: 5
9 Digite o elemento [1][2]: 6
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 2º Passo: Fazer uma função para contar o número de elementos pares de uma matriz.

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 2º Passo: Fazer uma função para contar o número de elementos pares de uma matriz.

```
1 ...
2 def contaPares(matriz):
3     """ Funcao que conta o numero de elementos pares em uma matriz
4     Parametro de entrada: list
5     Valor de retorno: int """
6
7     total = 0
8     for i in range(len(matriz)):
9         for j in range(len(matriz[0])):
10            if matriz[i][j] % 2 == 0:
11                total += 1
12    return total
13
14 def main():
15     """ Exercício Conta pares – programa principal """
16     numLinhas = int(input("Digite o numero de linhas : "))
17     numColunas = int(input("Digite o numero de colunas : "))
18     matriz = leMat(numLinhas, numColunas)
19     pares = contaPares(matriz)
20     print("O total de numeros pares e ", pares)
21
22 if __name__ == "__main__":
23     main()
```


Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 2º Passo: Fazer uma função para contar o número de elementos pares de uma matriz.

```
1 def main():
2     """ Exercício Conta pares – programa principal """
3     numLinhas = int(input("Digite o numero de linhas : "))
4     numColunas = int(input("Digite o numero de colunas : "))
5     matriz = leMat(numLinhas, numColunas)
6     pares = contaPares(matriz)
7     print("O total de numeros pares e ", pares)
8
9 if __name__ == "__main__":
10    main()
```

```
1 >>>
2 Digite o numero de linhas : 2
3 Digite o numero de colunas: 3
4 Digite o elemento [0][0]: 1
5 Digite o elemento [0][1]: 2
6 Digite o elemento [0][2]: 3
7 Digite o elemento [1][0]: 4
8 Digite o elemento [1][1]: 5
9 Digite o elemento [1][2]: 6
10
11 O total de numeros pares e 3
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 3º Passo: Como imprimir a matriz na tela ?

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 3º Passo: Como imprimir a matriz na tela ?

Imprimindo no formato de lista:

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 3º Passo: Como imprimir a matriz na tela ?

Imprimindo no formato de lista:

```
1 def main():
2     """ Exercício Conta pares – programa principal """
3     numLinhas = int(input("Digite o numero de linhas : "))
4     numColunas = int(input("Digite o numero de colunas : "))
5     matriz = leMat(numLinhas, numColunas)
6     pares = contaPares(matriz)
7     print("O total de numeros pares e ", pares)
8     print("A matriz lida foi: ", matriz)
9
10 if __name__ == "__main__":
11     main()
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

● 3º Passo: Como imprimir a matriz na tela ?

Imprimindo no formato de lista:

```
1 def main():
2     """ Exercício Conta pares – programa principal """
3     numLinhas = int(input("Digite o numero de linhas : "))
4     numColunas = int(input("Digite o numero de colunas : "))
5     matriz = leMat(numLinhas, numColunas)
6     pares = contaPares(matriz)
7     print("O total de numeros pares e ", pares)
8     print("A matriz lida foi : ", matriz)
9
10 if __name__ == "__main__":
11     main()
```

```
1 >>>
2 Digite o numero de linhas : 2
3 Digite o numero de colunas : 3
4 Digite o elemento [0][0]: 1
5 Digite o elemento [0][1]: 2
6 Digite o elemento [0][2]: 3
7 Digite o elemento [1][0]: 4
8 Digite o elemento [1][1]: 5
9 Digite o elemento [1][2]: 6
10
11 O total de numeros pares e 3
12 A matriz lida foi [[1,2,3],[4,5,6]]
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 3º Passo: Como imprimir a matriz na tela ?

Imprimindo no formato de matriz:

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

- 3º Passo: Como imprimir a matriz na tela ?

Imprimindo no formato de matriz:

```
1 def imprimirMatriz(matriz):
2     """Funcao para imprimir em formato de matriz
3     Parametro de entrada: list
4     Valor de retorno: sem saida"""
5
6     for i in range(len(matriz)):
7         for j in range(len(matriz[0])):
8             print(matriz[i][j], ' ') #um numero ao lado do outro
9         print() #pula linha
10    return
11
12 def main():
13     """Exercicio Conta pares - programa principal"""
14     numLinhas = int(input("Digite o numero de linhas : "))
15     numColunas = int(input("Digite o numero de colunas : "))
16     matriz = leMat(numLinhas, numColunas)
17     pares = contaPares(matriz)
18     print("O total de numeros pares e ", pares)
19     print("A matriz lida foi : ")
20     imprimirMatriz(matriz)
21
22 if __name__ == "__main__":
23     main()
```

Entrada e Saída de dados - Interação com o Usuário

Faça um programa que leia uma matriz digitada pelo usuário e conte quantos números pares existe na matriz, imprimindo o resultado na tela. Imprima a matriz na tela também.

● 3º Passo: Como imprimir a matriz na tela ?

Imprimindo no formato de matriz:

```
1 def imprimirMatriz(matriz):
2     """ Funcao para imprimir em formato de matriz
3     Parametro de entrada: list
4     Valor de retorno: sem saida"""
5
6     for i in range(len(matriz)):
7         for j in range(len(matriz[0])):
8             print matriz[i][j], #um numero ao lado do outro
9             print #pula linha
10    return
```

```
1 Digite o numero de linhas e colunas: 2, 3
2 Digite o elemento [0][0]: 1
3 Digite o elemento [0][1]: 2
4 Digite o elemento [0][2]: 3
5 Digite o elemento [1][0]: 4
6 Digite o elemento [1][1]: 5
7 Digite o elemento [1][2]: 6
8
9 O total de numeros pares e 3
10 A matriz lida foi:
11 1 2 3
12 4 5 6
```


Entrada e Saída de dados

Faça o programa do jogo: “Acerte os números”. Inicialmente, leia diversos números inteiros entrados pelo dono da casa de aposta, guardando numa lista. A leitura termina quando o dono digitar um número negativo. Em seguida o apostador poderá entrar com vários números para verificar o seu acerto.

Caso acerte um dos números da lista, informe do acerto e diga quantos números faltam acertar. O programa termina caso o apostador erre 10 vezes ou até ele acertar todos os números.

Obs.: caso o dono da casa de apostas tenha entrado com um número que já exista, ignore-o.

Autores

- **João C. P. da Silva** ▶ Lattes
- **Carla Delgado** ▶ Lattes
- **Ana Luisa Duboc** ▶ Lattes

Colaboradores

- **Anamaria Martins Moreira** ▶ Lattes
- **Fabio Mascarenhas** ▶ Lattes
- **Leonardo de Oliveira Carvalho** ▶ Lattes
- **Charles Figueiredo de Barros** ▶ Lattes
- **Fabrcio Firmino de Faria** ▶ Lattes

Computação I - Python

Aula 11 - Teórica: Entrada e Saída e Programa Principal

João C. P. da Silva

Carla A. D. M. Delgado

Ana Luisa Duboc

Dept. Ciência da Computação - UFRJ