

MAC 110 – Introdução à Ciência da Computação

Aula 21

Nelson Lago

BMAC – 2024



Exercícios

Exercício – soletrando

Escreva um programa que lê uma palavra e imprime a palavra “soletrada” (ou seja, com espaços entre as letras)



Exercício – soletrando

Escreva um programa que lê uma palavra e imprime a palavra “soletrada” (ou seja, com espaços entre as letras)

```
def main():
```

Exercício – soletrando

Escreva um programa que lê uma palavra e imprime a palavra “soletrada” (ou seja, com espaços entre as letras)

```
def main():  
    palavra = input("Digite a palavra: ")
```

Exercício – soletrando

Escreva um programa que lê uma palavra e imprime a palavra “soletrada” (ou seja, com espaços entre as letras)

```
def main():  
    palavra = input("Digite a palavra: ")  
    for letra in palavra:
```

Exercício – soletrando

Escreva um programa que lê uma palavra e imprime a palavra “soletrada” (ou seja, com espaços entre as letras)

```
def main():  
    palavra = input("Digite a palavra: ")  
    for letra in palavra:  
        print(letra, end=" ")
```

Exercício – soletrando

Escreva um programa que lê uma palavra e imprime a palavra “soletrada” (ou seja, com espaços entre as letras)

```
def main():  
    palavra = input("Digite a palavra: ")  
    for letra in palavra:  
        print(letra, end=" ")  
    print()
```


Exercício – elefantes

Escreva um programa que lê um inteiro positivo n e imprime a letra de “Um elefante incomoda muita gente...” n vezes



Exercício – elefantes

Escreva um programa que lê um inteiro positivo n e imprime a letra de “Um elefante incomoda muita gente...” n vezes

```
def main():  
    n = int(input("Quantas vezes? "))
```

Exercício – elefantes

Escreva um programa que lê um inteiro positivo n e imprime a letra de “Um elefante incomoda muita gente...” n vezes

```
def main():  
    n = int(input("Quantas vezes? "))  
    if n < 1:  
        return
```

Exercício – elefantes

Escreva um programa que lê um inteiro positivo n e imprime a letra de “Um elefante incomoda muita gente...” n vezes

```
def main():
    n = int(input("Quantas vezes? "))
    if n < 1:
        return
    print("Um elefante incomoda muita gente")
    print("Dois elefantes incomodam incomodam muito mais")
```

Exercício – elefantes

Escreva um programa que lê um inteiro positivo n e imprime a letra de “Um elefante incomoda muita gente...” n vezes

```
def main():
    n = int(input("Quantas vezes? "))
    if n < 1:
        return
    print("Um elefante incomoda muita gente")
    print("Dois elefantes incomodam incomodam muito mais")

    for i in range(
```

Exercício – elefantes

Escreva um programa que lê um inteiro positivo n e imprime a letra de “Um elefante incomoda muita gente...” n vezes

```
def main():
    n = int(input("Quantas vezes? "))
    if n < 1:
        return
    print("Um elefante incomoda muita gente")
    print("Dois elefantes incomodam incomodam muito mais")

    for i in range(2, n+1):
```

Exercício – elefantes

Escreva um programa que lê um inteiro positivo n e imprime a letra de “Um elefante incomoda muita gente...” n vezes

```
def main():
    n = int(input("Quantas vezes? "))
    if n < 1:
        return
    print("Um elefante incomoda muita gente")
    print("Dois elefantes incomodam incomodam muito mais")

    for i in range(2, n+1):
        print(i,
```

Exercício – elefantes

Escreva um programa que lê um inteiro positivo n e imprime a letra de “Um elefante incomoda muita gente...” n vezes

```
def main():
    n = int(input("Quantas vezes? "))
    if n < 1:
        return
    print("Um elefante incomoda muita gente")
    print("Dois elefantes incomodam incomodam muito mais")
    incomodam = "incomodam "
    for i in range(2, n+1):
        print(i,
```


Exercício – elefantes

Escreva um programa que lê um inteiro positivo n e imprime a letra de “Um elefante incomoda muita gente...” n vezes

```
def main():
    n = int(input("Quantas vezes? "))
    if n < 1:
        return
    print("Um elefante incomoda muita gente")
    print("Dois elefantes incomodam incomodam muito mais")
    incomodam = "incomodam "
    for i in range(2, n+1):
        print(i, incomodam * i)
```

Exercício – elefantes

Escreva um programa que lê um inteiro positivo n e imprime a letra de “Um elefante incomoda muita gente...” n vezes

```
def main():
    n = int(input("Quantas vezes? "))
    if n < 1:
        return
    print("Um elefante incomoda muita gente")
    print("Dois elefantes incomodam incomodam muito mais")
    incomodam = "incomodam "
    for i in range(2, n+1):
        print(i, "elefantes {}muita gente".format(incomodam * i))
```

Exercício – elefantes

Escreva um programa que lê um inteiro positivo n e imprime a letra de “Um elefante incomoda muita gente...” n vezes

```
def main():
    n = int(input("Quantas vezes? "))
    if n < 1:
        return
    print("Um elefante incomoda muita gente")
    print("Dois elefantes incomodam incomodam muito mais")
    incomodam = "incomodam "
    for i in range(2, n+1):
        print(i, "elefantes {}muita gente".format(incomodam * i))
        print(i + 1, "elefantes {}muito mais".format(incomodam * (i+1)))
```

Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.



Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):
```

Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):  
    lista = []
```

Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):  
    lista = []
```

```
    return lista
```

Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):  
    lista = []  
    s = s.lower()
```

```
    return lista
```


Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):  
    lista = []  
    s = s.lower()  
    início, quebra = 0, 0
```

```
    return lista
```


Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):  
    lista = []  
    s = s.lower()  
    início, quebra = 0, 0  
    while início < len(s):  
        quebra = len(s)  
  
    return lista
```

Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):
    lista = []
    s = s.lower()
    início, quebra = 0, 0
    while início < len(s):
        quebra = len(s)
        i = início
        while i < len(s):
            if s[i] == " ":
                quebra = i

            i += 1

    return lista
```

Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):  
    lista = []  
    s = s.lower()  
    início, quebra = 0, 0  
    while início < len(s):  
        quebra = len(s)  
        i = início  
        while i < len(s):  
            if s[i] == " ":  
                quebra = i  
                i = len(s)  
            i += 1  
  
    return lista
```

Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):  
    lista = []  
    s = s.lower()  
    início, quebra = 0, 0  
    while início < len(s):  
        quebra = len(s)  
        i = início  
        while i < len(s):  
            if s[i] == " ":  
                quebra = i  
                i = len(s)  
            i += 1  
        palavra = s[início:quebra]  
  
    return lista
```

Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):
    lista = []
    s = s.lower()
    início, quebra = 0, 0
    while início < len(s):
        quebra = len(s)
        i = início
        while i < len(s):
            if s[i] == " ":
                quebra = i
                i = len(s)
            i += 1
        palavra = s[início:quebra]
        if palavra[0] == 's':

    return lista
```

Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):
    lista = []
    s = s.lower()
    início, quebra = 0, 0
    while início < len(s):
        quebra = len(s)
        i = início
        while i < len(s):
            if s[i] == " ":
                quebra = i
                i = len(s)
            i += 1
        palavra = s[início:quebra]
        if palavra[0] == 's' and not palavra in lista:
            lista.append(palavra)

    return lista
```


Exercício — filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):
    lista = []
    s = s.lower()
    início, quebra = 0, 0
    while início < len(s):
        quebra = len(s)
        i = início
        while i < len(s):
            if s[i] == " ":
                quebra = i
                i = len(s)
            i += 1
        palavra = s[início:quebra]
        if palavra[0] == 's' and not palavra in lista:
            lista.append(palavra)
        início = quebra + 1
    return lista
```

Exercício – filtro de palavras

Escreva uma função que recebe uma string e devolve uma lista com todas as palavras da string que começam com a letra “s” (maiúscula ou minúscula). Palavras repetidas só devem aparecer uma vez.

```
def palavras_com_s(s):  
    lista = []  
    s = s.lower()  
    l = s.split()  
    for palavra in l:  
        if palavra[0] == 's' and not palavra in lista:  
            lista.append(palavra)  
    return lista
```

Exercício – gerador de anagramas

Escreva uma função que recebe uma string e devolve uma string que é um anagrama da string recebida.



Exercício – gerador de anagramas

Escreva uma função que recebe uma string e devolve uma string que é um anagrama da string recebida.

```
import random
```

Exercício – gerador de anagramas

Escreva uma função que recebe uma string e devolve uma string que é um anagrama da string recebida.

```
import random  
def anagrama(p):
```

Exercício – gerador de anagramas

Escreva uma função que recebe uma string e devolve uma string que é um anagrama da string recebida.

```
import random
def anagrama(p):
    ana = ""
```

Exercício – gerador de anagramas

Escreva uma função que recebe uma string e devolve uma string que é um anagrama da string recebida.

```
import random
def anagrama(p):
    ana = ""

    return ana
```

Exercício – gerador de anagramas

Escreva uma função que recebe uma string e devolve uma string que é um anagrama da string recebida.

```
import random
def anagrama(p):
    ana = ""

    i = random.randint(0, len(p) - 1)
    ana = ana + p[i]

    return ana
```


Exercício – gerador de anagramas

Escreva uma função que recebe uma string e devolve uma string que é um anagrama da string recebida.

```
import random
def anagrama(p):
    ana = ""
    while len(p) > 0:
        i = random.randint(0, len(p) - 1)
        ana = ana + p[i]

    return ana
```

Exercício – gerador de anagramas

Escreva uma função que recebe uma string e devolve uma string que é um anagrama da string recebida.

```
import random
def anagrama(p):
    ana = ""
    while len(p) > 0:
        i = random.randint(0, len(p) - 1)
        ana = ana + p[i]
        p =
    return ana
```

Exercício – gerador de anagramas

Escreva uma função que recebe uma string e devolve uma string que é um anagrama da string recebida.

```
import random
def anagrama(p):
    ana = ""
    while len(p) > 0:
        i = random.randint(0, len(p) - 1)
        ana = ana + p[i]
        p = p[0:i]
    return ana
```

Exercício – gerador de anagramas

Escreva uma função que recebe uma string e devolve uma string que é um anagrama da string recebida.

```
import random
def anagrama(p):
    ana = ""
    while len(p) > 0:
        i = random.randint(0, len(p) - 1)
        ana = ana + p[i]
        p = p[0:i] +
    return ana
```

Exercício – gerador de anagramas

Escreva uma função que recebe uma string e devolve uma string que é um anagrama da string recebida.

```
import random
def anagrama(p):
    ana = ""
    while len(p) > 0:
        i = random.randint(0, len(p) - 1)
        ana = ana + p[i]
        p = p[0:i] + p[i+1:]
    return ana
```

Exercício – somatório

Escreva uma função que recebe uma string com uma série de números separados por espaços e devolve a soma desses números



Exercício – somatório

Escreva uma função que recebe uma string com uma série de números separados por espaços e devolve a soma desses números

```
def somatório(s):
```

Exercício — somatório

Escreva uma função que recebe uma string com uma série de números separados por espaços e devolve a soma desses números

```
def somatório(s):  
    soma = 0
```


Exercício — somatório

Escreva uma função que recebe uma string com uma série de números separados por espaços e devolve a soma desses números

```
def somatório(s):  
    soma = 0  
  
    return soma
```

Exercício — somatório

Escreva uma função que recebe uma string com uma série de números separados por espaços e devolve a soma desses números

```
def somatório(s):  
    soma = 0  
  
    soma += int(n)  
    return soma
```

Exercício — somatório

Escreva uma função que recebe uma string com uma série de números separados por espaços e devolve a soma desses números

```
def somatório(s):  
    soma = 0  
    nums = s.split()  
  
    soma += int(n)  
    return soma
```

Exercício — somatório

Escreva uma função que recebe uma string com uma série de números separados por espaços e devolve a soma desses números

```
def somatório(s):  
    soma = 0  
    nums = s.split()  
    for n in nums:  
        soma += int(n)  
    return soma
```

Previously on MAC110...

```
arq = open("arquivo.txt", "r")  
print(arq.read())  
arq.close()
```

```
arq = open("arquivo.txt", "r")
print(arq.read())
arq.close()
```

```
arq = open("arquivo.txt", "r")
linha = arq.readline()
while linha != "":
    ...
    linha = arq.readline()
arq.close()
```

```
arq = open("arquivo.txt", "r")
print(arq.read())
arq.close()
```

```
arq = open("arquivo.txt", "r")
linha = arq.readline()
while linha != "":
    ...
    linha = arq.readline()
arq.close()
```

```
arq = open("arquivo.txt", "r")
for linha in arq:
    ...
arq.close()
```


Arquivos

```
arq = open("arquivo.txt", "r")
print(arq.read())
arq.close()
```

```
arq = open("arquivo.txt", "r")
linha = arq.readline()
while linha != "":
    ...
    linha = arq.readline()
arq.close()
```

```
arq = open("arquivo.txt", "r")
for linha in arq:
    ...
arq.close()
```

```
arq = open("arquivo.txt", "r")
linhas = arq.readlines()
arq.close()
for linha in linhas:
    ...
```

```
poeminha = [ "Batatinha quando nasce",  
             "Espalha a rama pelo chão",  
             "A menina quando dorme",  
             "Põe a mão no coração" ]  
  
arq = open("arquivo.txt", "w")  
for linha in poeminha:  
    arq.write(linha + "\n")  
arq.close()
```

```
poeminha = [ "Batatinha quando nasce",  
             "Espalha a rama pelo chão",  
             "A menina quando dorme",  
             "Põe a mão no coração" ]  
  
arq = open("arquivo.txt", "w")  
for linha in poeminha:  
    arq.write(linha + "\n")  
arq.close()
```

```
poeminha = [ "Batatinha quando nasce",  
             "Espalha a rama pelo chão",  
             "A menina quando dorme",  
             "Põe a mão no coração" ]  
  
arq = open("arquivo.txt", "w")  
arq.write("\n".join(poeminha))  
arq.close()
```

Arquivos

```
poeminha = [ "Batatinha quando nasce",  
             "Espalha a rama pelo chão",  
             "A menina quando dorme",  
             "Põe a mão no coração" ]  
  
arq = open("arquivo.txt", "w")  
for linha in poeminha:  
    arq.write(linha + "\n")  
arq.close()
```

```
poeminha = [ "Batatinha quando nasce",  
             "Espalha a rama pelo chão",  
             "A menina quando dorme",  
             "Põe a mão no coração" ]  
  
arq = open("arquivo.txt", "w")  
arq.write("\n".join(poeminha))  
arq.close()
```

```
poeminha = "Batatinha quando nasce\n" \  
           "Espalha a rama pelo chão\n" \  
           "A menina quando dorme\n" \  
           "Põe a mão no coração"  
  
arq = open("arquivo.txt", "w")  
arq.write(poeminha)  
arq.close()
```

Arquivos

```
poeminha = [ "Batatinha quando nasce",  
             "Espalha a rama pelo chão",  
             "A menina quando dorme",  
             "Põe a mão no coração" ]  
  
arq = open("arquivo.txt", "w")  
for linha in poeminha:  
    arq.write(linha + "\n")  
arq.close()
```

```
poeminha = [ "Batatinha quando nasce",  
             "Espalha a rama pelo chão",  
             "A menina quando dorme",  
             "Põe a mão no coração" ]  
  
arq = open("arquivo.txt", "w")  
arq.write("\n".join(poeminha))  
arq.close()
```

```
poeminha = "Batatinha quando nasce\n" \  
           "Espalha a rama pelo chão\n" \  
           "A menina quando dorme\n" \  
           "Põe a mão no coração"  
  
arq = open("arquivo.txt", "w")  
arq.write(poeminha)  
arq.close()
```

```
poeminha = ("Batatinha quando nasce\n"  
           "Espalha a rama pelo chão\n"  
           "A menina quando dorme\n"  
           "Põe a mão no coração")  
  
arq = open("arquivo.txt", "w")  
for linha in poeminha.split("\n"):  
    arq.write(linha + "\n")  
arq.close()
```

Exercícios

Exercício – somatório II

Escreva um programa que lê um arquivo em que cada linha contém números separados por espaços e imprime (1) a soma dos números de cada linha e (2) a soma de todos os números



Exercício — somatório II

Escreva um programa que lê um arquivo em que cada linha contém números separados por espaços e imprime (1) a soma dos números de cada linha e (2) a soma de todos os números

```
def main():
```


Exercício — somatório II

Escreva um programa que lê um arquivo em que cada linha contém números separados por espaços e imprime (1) a soma dos números de cada linha e (2) a soma de todos os números

```
def main():
```

```
    print("Soma total:", total)
```

Exercício – somatório II

Escreva um programa que lê um arquivo em que cada linha contém números separados por espaços e imprime (1) a soma dos números de cada linha e (2) a soma de todos os números

```
def main():  
  
    total = 0  
  
    print("Soma total:", total)
```

Exercício — somatório II

Escreva um programa que lê um arquivo em que cada linha contém números separados por espaços e imprime (1) a soma dos números de cada linha e (2) a soma de todos os números

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    total = 0

    print("Soma total:", total)
```

Exercício — somatório II

Escreva um programa que lê um arquivo em que cada linha contém números separados por espaços e imprime (1) a soma dos números de cada linha e (2) a soma de todos os números

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    total = 0
    for linha in arq:

    print("Soma total:", total)
```

Exercício — somatório II

Escreva um programa que lê um arquivo em que cada linha contém números separados por espaços e imprime (1) a soma dos números de cada linha e (2) a soma de todos os números

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    total = 0
    for linha in arq:
        val = somatório(linha)
        print("Soma:", val)
        total += val
    print("Soma total:", total)
```

Exercício – matriz

Escreva um programa que lê um arquivo de texto em que cada linha contém números separados por espaços e transforma o conteúdo desse arquivo em uma matriz



Exercício – matriz

Escreva um programa que lê um arquivo de texto em que cada linha contém números separados por espaços e transforma o conteúdo desse arquivo em uma matriz

```
def main():
```

Exercício — matriz

Escreva um programa que lê um arquivo de texto em que cada linha contém números separados por espaços e transforma o conteúdo desse arquivo em uma matriz

```
def main():  
    nome_arq = input("Qual o arquivo? ")  
    arq = open(nome_arq, "r")
```


Exercício — matriz

Escreva um programa que lê um arquivo de texto em que cada linha contém números separados por espaços e transforma o conteúdo desse arquivo em uma matriz

```
def main():  
    nome_arq = input("Qual o arquivo? ")  
    arq = open(nome_arq, "r")  
    matriz = []
```

Exercício — matriz

Escreva um programa que lê um arquivo de texto em que cada linha contém números separados por espaços e transforma o conteúdo desse arquivo em uma matriz

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    matriz = []
    for linha in arq:
```

Exercício — matriz

Escreva um programa que lê um arquivo de texto em que cada linha contém números separados por espaços e transforma o conteúdo desse arquivo em uma matriz

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    matriz = []
    for linha in arq:
        l = linha.split()
```

Exercício — matriz

Escreva um programa que lê um arquivo de texto em que cada linha contém números separados por espaços e transforma o conteúdo desse arquivo em uma matriz

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    matriz = []
    for linha in arq:
        l = linha.split()
        for i in range(len(l)):
            l[i] = int(l[i])
```

Exercício — matriz

Escreva um programa que lê um arquivo de texto em que cada linha contém números separados por espaços e transforma o conteúdo desse arquivo em uma matriz

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    matriz = []
    for linha in arq:
        l = linha.split()
        for i in range(len(l)):
            l[i] = int(l[i])
        matriz.append(l)
```

Exercício – cálculo de notas

Escreva um programa que leia um arquivo CSV onde cada linha contém o nome e as notas de um estudante e, para cada estudante, calcule e imprima a média das notas. Ao final, o programa deve imprimir também a média da turma.

Exercício — cálculo de notas

Exemplo: para o arquivo notas.csv com:

```
José Augusto, 10, 15, 20, 30, 40
Pedro Mé,      23, 16, 19, 22
Suzana Ré,     8, 22, 17, 14, 32, 17, 24, 21, 2, 9, 11, 17
Gisela Fox,    12, 28, 21, 45, 26, 10
João Mendonça, 14, 32, 25, 16, 89
```

a saída deve ser:

```
Digite o nome do arquivo: notas.csv
```

```
Aluno: José Augusto Média = 23.0
Aluno: Pedro Mé Média = 20.0
Aluno: Suzana Ré Média = 16.1
Aluno: Gisela Fox Média = 23.6
Aluno: João Mendonça Média = 35.2
```

```
Média da turma = 23.6
```

```
Dica para formatação: usar aluno.ljust(13)
```

Exercício – cálculo de notas

Exercício – cálculo de notas

```
def main():  
    nome_arq = input("Qual o arquivo? ")  
    arq = open(nome_arq, "r")
```

Exercício – cálculo de notas

```
def main():  
    nome_arq = input("Qual o arquivo? ")  
    arq = open(nome_arq, "r")
```

```
arq.close()
```

Exercício – cálculo de notas

```
def main():  
    nome_arq = input("Qual o arquivo? ")  
    arq = open(nome_arq, "r")  
    n_alunos = 0  
    soma_geral = 0
```

```
arq.close()
```

Exercício – cálculo de notas

```
def main():  
    nome_arq = input("Qual o arquivo? ")  
    arq = open(nome_arq, "r")  
    n_alunos = 0  
    soma_geral = 0  
    for linha in arq:
```

```
        arq.close()
```

Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
```

```
arq.close()
```

Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0]
```

```
arq.close()
```

Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0].strip()
```

```
arq.close()
```

Exercício – cálculo de notas

```
def main():  
    nome_arq = input("Qual o arquivo? ")  
    arq = open(nome_arq, "r")  
    n_alunos = 0  
    soma_geral = 0  
    for linha in arq:  
        itens = linha.split(",")  
        nome = itens[0].strip()  
        notas = itens[1:]
```

```
arq.close()
```


Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0].strip()
        notas = itens[1:]
        média = 0
        for val in notas:
            média +=
```

```
arq.close()
```

Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0].strip()
        notas = itens[1:]
        média = 0
        for val in notas:
            média += float(val)
```

```
arq.close()
```

Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0].strip()
        notas = itens[1:]
        média = 0
        for val in notas:
            média += float(val)
        média /= len(notas)
        print("Aluno:",
            nome, "Média:", média)
    arq.close()
```

Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0].strip()
        notas = itens[1:]
        média = 0
        for val in notas:
            média += float(val)
        média /= len(notas)
        print("Aluno:", nome

arq.close()
```

Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0].strip()
        notas = itens[1:]
        média = 0
        for val in notas:
            média += float(val)
        média /= len(notas)
        print("Aluno:", nome.ljust(13),

arq.close()
```

Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0].strip()
        notas = itens[1:]
        média = 0
        for val in notas:
            média += float(val)
        média /= len(notas)
        print("Aluno:", nome.ljust(13), "Média =",          média )

    arq.close()
```

Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0].strip()
        notas = itens[1:]
        média = 0
        for val in notas:
            média += float(val)
        média /= len(notas)
        print("Aluno:", nome.ljust(13), "Média =", "{:.1f}".format(média))

    arq.close()
```

Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0].strip()
        notas = itens[1:]
        média = 0
        for val in notas:
            média += float(val)
        média /= len(notas)
        print("Aluno:", nome.ljust(13), "Média =", "{:.1f}".format(média))
        soma_geral += média
        n_alunos += 1

    arq.close()
```


Exercício – cálculo de notas

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0].strip()
        notas = itens[1:]
        média = 0
        for val in notas:
            média += float(val)
        média /= len(notas)
        print("Aluno:", nome.ljust(13), "Média =", "{:.1f}".format(média))
        soma_geral += média
        n_alunos += 1
    print("Média da turma = {:.1f}".format(soma_geral / n_alunos))
    arq.close()
```

Exercício – cálculo de notas

Modifique o programa anterior para que os resultados sejam salvos em um arquivo ao invés de impressos na tela

Exercício — cálculo de notas

Modifique o programa anterior para que os resultados sejam salvos em um arquivo ao invés de impressos na tela

```
def main():
    nome_arq = input("Qual o arquivo? ")
    arq = open(nome_arq, "r")
    saída = open("sai.txt", "w")
    n_alunos = 0
    soma_geral = 0
    for linha in arq:
        itens = linha.split(",")
        nome = itens[0].strip()
        notas = itens[1:]
        média = 0
        for val in notas:
            média += float(val)
        média /= len(notas)
        saída.write("Aluno: " + nome.ljust(14) + "Média = " + str(round(média,1)) + "\n")
        soma_geral += média
        n_alunos += 1
    saída.write("Média da turma = {:.1f}\n".format(soma_geral / n_alunos))
    arq.close()
    saída.close()
```