

Exercícios de Fixação – Python (Cap. 5)

NOME: Camilo Bustamante Moreira

DISCIPLINA: Algoritmos

PROFESSOR: Msc. Lucas Gonçalves Nadalete

Instruções de Entrega:

- **Atividade individual;**
- Entrega a ser realizada até o dia **30/03/2016 às 23:55 hrs.**
- A entrega deve ser feita em um arquivo **.PDF** contendo o nome da disciplina, nome do professor, nome do aluno, e os respectivos enunciados e soluções dos exercícios em sequência;
- Enviar o arquivo para o email lucas.nadalete@fatec.sp.gov.br;
- O título do email deve ser: **<nome_aluno> - Capítulo 5 - Algoritmos** (para evitar erros de digitação, copie/cole o título substituindo apenas o nome).

5.1) Modifique o programa para exibir os números de 1 a 100.

Listagem 5.5 – Imprimindo de 1 a 3 com while

```
x=1 ❶  
while x<=3: ❷  
    print(x) ❸  
    x = x + 1 ❹
```

```
x = 1  
while x <= 100:  
    print(x)  
    x = x + 1
```

5.2) Modifique o programa para exibir os números de 50 a 100.

Listagem 5.5 – Imprimindo de 1 a 3 com while

```
x=1 ❶  
while x<=3: ❷  
    print(x) ❸  
    x = x + 1 ❹
```

```
x = 50  
while x <= 100:  
    print(x)  
    x = x + 1
```

5.3) Escreva um programa para escrever a contagem regressiva do lançamento de um foguete. O programa deve imprimir 10, 9, 8, ..., 1, 0 e Fogo! na tela.

```
x = 10  
while x >= 0:  
    print(x)  
    if(x == 0):  
        print("Fogo!")  
    x = x - 1
```

5.4) Modifique o programa para imprimir de 1 até o número digitado pelo usuário, mas dessa vez apenas os números ímpares.

Listagem 5.8 – Impressão de números pares de 0 até um número digitado pelo usuário, sem if

```
fim=int(input("Digite o último número a imprimir:"))
x = 0
while x <= fim:
    print(x)
    x = x + 2
```

```
fim = int(input("Digite o último número a imprimir: "))
x = 0
while x <= fim:
    if(x % 2 != 0):
        print(x)
    x = x + 1
```

5.5) Reescreva o programa anterior para escrever os 10 primeiros múltiplos de 3.

```
x = 3
mult3 = 0
while mult3 < 10:
    if(x % 3 == 0):
        print(x)
        mult3 += 1
    x = x + 1
```

5.6) Altere o programa a seguir para exibir os resultados no mesmo formato de uma tabuada: 2x1 = 2, 2x2 = 4, ...

Listagem 5.9 – Tabuada simples

```
n = int(input("Tabuada de:"))
x = 1
while x <=10:
    print(n+x)
    x=x+1
```

```
n = int(input("Tabuada de: "))
x = 0
while x <= 10:
    print("%i x %i = %i" %(n, x, (n * x)))
    x = x + 1
```

5.7) Modifique o programa anterior de forma que o usuário também digite o início e o fim da tabuada em vez de começar com 1 e 10.

```
n = int(input("Tabuada de: "))
x = int(input("Começando em: "))
y = int(input("Terminando em: "))
while x <= y:
    print("%i x %i = %i" %(n, x, (n * x)))
    x = x + 1
```

5.8) Escreva um programa que leia dois números. Imprima o resultado da multiplicação do primeiro pelo segundo. Utilize apenas os operadores de soma e subtração para calcular o resultado. Lembre-se de que podemos entender a multiplicação de dois números como somas sucessivas de um deles. Assim, 4x5 = 5+5+5+5 = 4+4+4+4+4.

```
num1 = int(input("Entre com o multiplicando: "))
num2 = int(input("Entre com o multiplicador: "))
x = 0
result = 0
```

```

while(x < num2):
    result += num1
    x += 1
print(result)

```

5.9) Escreva um programa que leia dois números. Imprima a divisão inteira do primeiro pelo segundo, assim como o resto da divisão. Utilize apenas os operadores de soma e subtração para calcular o resultado. Lembre-se de que podemos entender o quociente da divisão de dois números como a quantidade de vezes que podemos retirar o divisor do dividendo. Logo, $20/4 = 5$, uma vez que podemos subtrair 4 cinco vezes de 20.

```

num1 = int(input("Entre com o dividendo: "))
num2 = int(input("Entre com o divisor: "))
x = 0
divisao = num1
resto = 0
result = 0
while(divisao > 0):
    if(divisao > num2):
        divisao -= num2
        result += 1
    else:
        resto = divisao
        divisao -= divisao
print("Divisao: %i \nResto: %i" %(result, resto))

```

5.10) Modifique o programa da listagem 5.10 para que aceite respostas com letras maiúsculas e minúsculas em todas as questões.

Listagem 5.10 – Contagem de questões corretas

```

pontos=0
questão = 1
while questão <= 3:
    resposta = input("Resposta da questão %d: " % questão)
    if questão==1 and resposta == "b":
        pontos = pontos + 1
    if questão==2 and resposta == "a":
        pontos = pontos + 1
    if questão==3 and resposta == "d":
        pontos = pontos + 1
    questão +=1
print("O aluno fez %d ponto(s)" % pontos)

```

```

pontos = 0
questao = 1
while questao <= 3:
    resposta = input("Resposta da questão %i: " %questao)
    if ((questao == 1) and (resposta == "b" or resposta == "B")):
        pontos = pontos + 1
    if ((questao == 2) and (resposta == "a" or resposta == "A")):
        pontos = pontos + 1
    if ((questao == 3) and (resposta == "d" or resposta == "D")):
        pontos = pontos + 1
    questao += 1
print("O aluno fez %i pontos" %pontos)

```

5.11) Escreva um programa que pergunte o depósito inicial e a taxa de juros de uma poupança. Exiba os valores mês a mês para os 24 primeiros meses. Escreva o total ganho com juros no período.

```

dep = float(input("Deposito: "))
juros = float(input("Juros: "))

```

```

total = dep
if juros < 10:
    juros = juros * .01
elif juros > 10 and juros < 100:
    juros = juros * .1
x = 1
while x < 24:
    print("Até o mês %i você já tem um total de R$%.2f" %(x, total))
    total += (dep * juros)
    x += 1
print("O total ganho ao final do %iª mês foi R$%.2f" %(x, total))

```

5.12) Altere o programa anterior de forma a perguntar também o valor depositado mensalmente. Esse valor será depositado no início de cada mês e você deve considerá-lo para o cálculo de juros do mês seguinte.

```

dep = float(input("Deposito: "))
juros = float(input("Juros: "))
total = dep
if juros < 10:
    juros = juros * .01
elif juros > 10 and juros < 100:
    juros = juros * .1
x = 1
while x < 24:
    dep = float(input("Deposito para o %iº mês : " %x))
    total += dep
    print("Até o mês %i você já tem um total de R$%.2f" %(x, total))
    total += (dep * juros)
    x += 1
print("O total ganho ao final do %iª mês foi R$%.2f" %(x, total))

```

5.13) Escreva um programa que pergunte o valor inicial de uma dívida e o juro mensal. Pergunte também o valor mensal que será pago. Imprima o número de meses para que a dívida seja paga, o total pago e o total de juros pago.

```

divi = float(input("Dívida: "))
juros = float(input("Juros: "))
pag = float(input("Pagamento mensal: "))
total = divi
if juros < 10:
    juros = juros * .01
elif juros > 10 and juros < 100:
    juros = juros * .1
x = 0
jurosT = total
diviIni = divi
while divi > 0:
    divi = divi + (divi * juros) - pag
    total = total + (total * juros)
    x += 1
jurosT = total - diviIni
print("Meses para pagar a dívida: %i \nTotal pago: R$%.2f \nTotal de juros pago: R$%.2f" %(x, total, jurosT))

```

5.14) Escreva um programa que leia números inteiros do teclado. O programa deve ler os números até que o usuário digite 0 (zero). No final da execução exiba a quantidade de números digitados, assim como a soma e a média aritmética.

```

num = 0
media = 0
soma = 0

```

```

x = 0
while(True):
    num = int(input("Informe um número: "))
    if num == 0:
        break
    print("Número computado.")
    x += 1
    soma += num
media = soma/x
print("Soma: %i \nMédia aritmética: %i" %(soma, media))

```

5.15) Escreva um programa para controlar uma pequena máquina registradora. Você deve solicitar ao usuário que digite o código do produto e a quantidade comprada. Utilize a tabela de códigos abaixo para obter o preço de cada produto:

Código	Preço
1	0,50
2	1,00
3	4,00
5	7,00
9	8,00

Seu programa deve exibir o total das compras depois que o usuário digitar 0. Qualquer outro código deve gerar a mensagem de erro "Código inválido".

```

preco = 0
while(True):
    codProd = int(input("Informe o código do produto: "))
    if codProd == 1:
        qtde = int(input("Informe a quantidade comprada: "))
        preco += qtde * .5
    elif codProd == 2:
        qtde = int(input("Informe a quantidade comprada: "))
        preco += qtde * 1
    elif codProd == 3:
        qtde = int(input("Informe a quantidade comprada: "))
        preco += qtde * 4
    elif codProd == 5:
        qtde = int(input("Informe a quantidade comprada: "))
        preco += qtde * 7
    elif codProd == 9:
        qtde = int(input("Informe a quantidade comprada: "))
        preco += qtde * 8
    elif codProd == 0:
        break
    else:
        print("Código inválido.")
print("Preço final: R$%.2f" %preco)

```

5.16) Execute o programa (Listagem 5.14) para os seguintes valores: 501, 745, 384, 2, 7 e 1.

Listagem 5.14 – Contagem de cédulas

```
valor=int(input("Digite o valor a pagar:"))
cédulas=0
atual=50
apagar=valor
while True:
    if atual<=apagar:
        apagar-=atual
        cédulas+=1
    else:
        print("%d cédula(s) de R${}" % (cédulas, atual))
        if apagar == 0:
            break
        if atual == 50:
            atual = 20
        elif atual == 20:
            atual = 10
        elif atual == 10:
            atual = 5
        elif atual == 5:
            atual = 1
        cédulas = 0
```

5.17) O que acontece se digitarmos 0 (zero) no valor a pagar? Vai retornar na tela “o cédula(s) de R\$50”;

5.18) Modifique o programa para também trabalhar com notas de R\$100.

```
valor = int(input("Digite o valor a pagar: "))
cedulas = 0
atual = 100
apagar = valor
while True:
    if atual <= apagar:
        apagar -= atual
        cedulas += 1
    else:
        print("%i cédula(s) de R${}" %(cedulas, atual))
        if apagar == 0:
            break
        if atual == 100:
            atual = 50
        elif atual == 50:
            atual = 20
        elif atual == 20:
            atual = 10
        elif atual == 10:
            atual = 5
        elif atual == 5:
            atual = 1
        cedulas = 0
```

5.19) Modifique o programa para aceitar valores decimais, ou seja, também contar moedas de 0,01; 0,02; 0,05; 0,10 e 0,50.

```
valor = float(input("Digite o valor a pagar: "))
cedulas = 0
atual = 100
apagar = valor
while True:
    if atual <= apagar:
        apagar -= atual
```

```

    cédulas += 1
else:
    print("%i cédula(s) de R$%.2f" %(cédulas, atual))
    if apagar < .01:
        print("O restante a ser pago é igual a %.10f" %apagar)
        break
    if atual == 100:
        atual = 50
    elif atual == 50:
        atual = 20
    elif atual == 20:
        atual = 10
    elif atual == 10:
        atual = 5
    elif atual == 5:
        atual = 1
    elif atual == 1:
        atual = .5
    elif atual == .5:
        atual = .1
    elif atual == .1:
        atual = .05
    elif atual == .05:
        atual = .02
    elif atual == .02:
        atual = .01
    cédulas = 0

```

5.20) O que acontece se digitarmos 0,001 no programa anterior? Caso ele não funcione, altere-o de forma a corrigir o problema.

```

cédulas = 0
atual = 100
apagar = valor
while True:
    if atual <= apagar:
        apagar -= atual
        cédulas += 1
    else:
        print("%i cédula(s) de R$%.2f" %(cédulas, atual))
        if apagar < .01:
            print("O restante a ser pago é igual a %.10f" %apagar)
            break
        if atual == 100:
            atual = 50
        elif atual == 50:
            atual = 20
        elif atual == 20:
            atual = 10
        elif atual == 10:
            atual = 5
        elif atual == 5:
            atual = 1
        elif atual == 1:
            atual = .5
        elif atual == .5:
            atual = .1
        elif atual == .1:
            atual = .05
        elif atual == .05:
            atual = .02
        elif atual == .02:
            atual = .01
        cédulas = 0

```

5.21) Reescreva o programa da listagem 5.14, de forma a continuar executando até que o valor digitado seja 0. Utilize repetições aninhadas.

```
while True:
    valor = int(input("Digite o valor a pagar: "))
    if valor == 0:
        break
    cedulas = 0
    atual = 50
    apagar = valor
    while True:
        if atual <= apagar:
            apagar -= atual
            cedulas += 1
        else:
            print("%i cédula(s) de R$%i" %(cedulas, atual))
            if apagar == 0:
                break
            elif atual == 50:
                atual = 20
            elif atual == 20:
                atual = 10
            elif atual == 10:
                atual = 5
            elif atual == 5:
                atual = 1
            cedulas = 0
```

5.22) Escreva um programa que exiba uma lista de opções (menu): adição, subtração, divisão, multiplicação e sair. Imprima a tabuada da operação escolhida. Repita até que a opção saída seja escolhida.

```
while(True):
    op = input("Informe o número ou o simbolo da operação desejada: \n1 - soma (+); \n2 - subtração (-); \n3 - multiplicação (*); \n4 - divisão (/); \n5 - sair. \n").lower()
    a = 0
    if(op == "1" or op == "+"):
        while a <= 10:
            b = 0
            while b <= 10:
                print(" %i + %i é: %i" %(a, b, a + b))
                b += 1
            print("\n")
            a += 1
        break
    elif(op == "2" or op == "-"):
        while a <= 10:
            b = 0
            while b <= 10:
                print(" %i - %i é: %i" % (a, b, a - b))
                b += 1
            print("\n")
            a += 1
        break
    elif(op == "3" or op == "*"):
        while a <= 10:
            b = 0
            while b <= 10:
                print(" %i * %i é: %i" % (a, b, a * b))
                b += 1
            print("\n")
```



```

        a += 1
    break
elif(op == "4" or op == "/"):
    while a <= 10:
        b = 0
        while b <= 10:
            print(" %i / %i é: %i" % (a, b, a / b))
            b += 1
        print("\n")
        a += 1
    break
elif(op == "5" or op == "sair"):
    break
else:
    print("Operação inválida.")

```

5.23) Escreva um programa que leia um número e verifique se é ou não um número primo. Para fazer essa verificação, calcule o resto da divisão do número por 2 e depois por todos os números ímpares até o número lido. Se o resto de uma dessas divisões for igual a zero, o número não é primo. Observe que 0 e 1 não são primos e que 2 é o único número primo que é par.

```

num = int(input("Informe um numero: "))
if num == 2 or num == 3:
    print("É primo.")
elif ((num == 0) or (num == 1) or (num % 2 == 0)):
    print("Não é primo.")
else:
    x = 3
    while x < num:
        resto = num % x
        x += 2
        if resto == 0:
            print("Não é primo.")
        else:
            print("É primo.")
            break

```

5.24) Modifique o programa anterior de forma a ler um número n. Imprima os n primeiros números primos.

5.25) Escreva um programa que calcule a raiz quadrada de um número. Utilize o método de Newton para obter um resultado aproximado. Sendo n o número a obter a raiz quadrada, considere a base b=2. Calcule p usando a fórmula $p = (b + (b/n))/2$. Agora, calcule o quadrado de p. A cada passo, faça $b = p$ e recalcule p usando a fórmula apresentada. Pare quando a diferença absoluta entre n e o quadrado de p for menor que 0,0001.

5.26) Escreva um programa que calcule o resto da divisão inteira entre dois números. Utilize apenas as operações de soma e subtração para calcular este resultado.

5.27) Escreva um programa que verifique se um número é palíndromo. Um número é palíndromo se continua o mesmo caso seus dígitos sejam invertidos. Exemplos: 454, 10501.

```

num = int(input("Entre com um número: "))
num = str(num)
x = 0
palind = ""
while x <= len(num) - 1:
    palind = num[x] + palind #Inverter o número. Mudar o último dígito para o começo.

```

```
    x += 1
if(num == palind)
    print("É um palindromo.")
else:
    print("Não é um palindromo.")
```