



www.estudar.com.vc

Resumo e Exercícios P1

MAC2166 Em Python





Resumo

Estrutura básica de um programa

```
def main():  
    # Comandos  
    print("Hello World!") # indentação é essencial  
  
# Executa o programa  
main()
```

Variáveis e Tipos

```
# Tipos básicos de variáveis (do # pra frente na linha é um comentário)  
inteiro = 3          # int  
decimal = 3.2       # float  
string = "Ola!"    # str  
boolean = (inteiro == 3) # bool (True nesse caso)
```

Operadores

```
a = 3 + 5 # a == 8 Soma  
b = 7 - 5 # b == 2 Subtração  
c = a * b # c == 16 Multiplicação  
d = c / 5 # d == 3.2 Divisão  
e = c // 5 # e == 3 Divisão Inteira  
f = c % 5 # f == 1 Resto da divisão inteira  
g = 2 ** 3 # g == 8 Potência
```

print() e input()

```
nome = input("Digite seu nome: ") # Lê uma string  
num = int(input("Digite um inteiro: ")) # Lê uma string e converte para inteiro  
dec = float(input("Digite um decimal: ")) # Lê uma string e converte para float  
boolean = bool(input("Digite um booleano: ")) # Não funcionaria como você quer,  
qualquer string não vazia seria convertida para True, mesmo que o usuário digitasse  
'False'  
  
print("Ola") # Imprime 'Ola' na saída padrão  
print(f) # 1  
print("O valor de g e:", g) # 'O valor de num e: 8'  
print("O valor de num e: %d" % (g)) # Idem
```



Condicionais e Loops

```
a = 5
if a > 6:      # condição é bool, executa se True
    a = a + 2
elif a > 3:   # Executa esse
    a = a - 2
else:        # Executa se todos os outros if/elif forem False
    a = 0
```

```
i = 5
while i > 0:   # Executa o bloco enquanto a condição for True
    print("Ola!") # Imprime 'Ola!' 5 vezes
    i = i - 1
```

Funções

```
def nome_da_funcao(parametro1):
    # comandos
    print("Ola!", parametro1)

def main(): # Também é uma função!
    nome_da_funcao("Bem-vindo") # Imprime 'Ola! Bem-vindo'

main()
```



Exercícios de Fixação

1. Pedra, Papel ou Tesoura

Internet

Faça um programa que pede 2 entradas, cada uma podendo ser “pedra”, “papel” ou “tesoura” e indique o vencedor.

2. Soma de Sequência

Internet

Dada uma sequência de números inteiros diferentes de zero, terminada por um zero, calcular a sua soma. Por exemplo, para a sequência:

12 17 4 -6 8 0

Seu programa deve escrever 35.



Exercícios de Prova

1. Simulação de Código

P1 2018 – Questão 1

Simule o código abaixo e determine as saídas impressas do programa

```
def main():
    k = 25
    a = 0
    while k >= a:
        b = k // 2
        if b * 2 == k:
            a = a + 3
        else:
            a = a + 2
        k = k - 3
        x = a % 7
        print(x * 7 + 14)
main()
```

2. Expressões Lógicas

P1 2018 – Questão 2

Para cada item envolvendo a análise de expressões lógicas, selecione as opções corretas (desde nenhuma até todas opções podem ser aplicáveis).

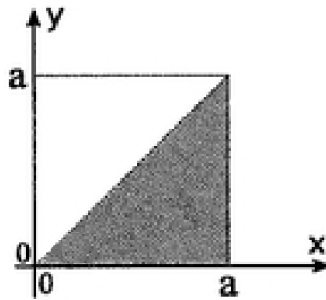
a. Dadas duas variáveis inteiras a e b , selecione as expressões equivalentes a:

$$a \geq b$$

- A.** $a < b == False$ **D.** $a > b + 1$ **G.** $a > b$ and $a == b$
B. $not(a \leq b$ or $a! = b)$ **E.** $not(a < b)$ **H.** $not(a \leq b$ and $a! = b)$
C. $(a // 2) \geq (b // 2)$ **F.** $a > b$ or $a == b$ **I.** $a > b - 1$



b. Dada as coordenadas reais x e y de um ponto, selecione todas as expressões lógicas verdadeiras para o caso em que o ponto pertença à região sombreada da figura, e falsas para o caso contrário. A região sombreada não inclui as linhas de fronteira.



- A.** $x < a$ and $y > 0$ and $x > y$
- B.** $\text{not}(x \geq a$ and $y \leq 0$ and $x \leq y)$
- C.** $x < a$ or $y > 0$ or $x > y$
- D.** $y < x$ and $\text{not}(x \geq a$ or $0 \geq y)$
- E.** $x < a$ or $(0 < y$ and $y < x)$
- F.** $\text{not}(x \geq a$ or $y \leq 0$ or $x \leq y)$
- G.** $x > 0$ or $x < a$ or $y < a$ or $y > 0$ or $x > y$
- H.** $x \geq a$ or $y \leq 0$ or $x \leq y$
- I.** $x < a$ and $y > 0$ and $\text{not}(x \leq y)$
- J.** $x > 0$ and $x < a$ and $y < a$ and $y > 0$ and $x > y$

3. Laços e Funções

P1 2016 – Questão 3

Escreva um programa que sorteia um inteiro secreto ≥ 0 , com 3 dígitos, que deverá ser adivinhado por um jogador. Zeros à esquerda são considerados.



Em cada **tentativa** do jogador para adivinhar o inteiro secreto, o programa:

- lê um inteiro “chute” digitado pelo jogador, $\text{chute} \geq 0$, com 3 dígitos, zeros à esquerda são considerados;
- imprime o número de **posições certas** entre inteiro secreto e chute;
e
- imprime o número de **dígitos casados** entre inteiro secreto e chute.

O programa deve parar assim que o número secreto for adivinhado pelo jogador ou depois de 6 tentativas frustradas. Ao final, o programa deve imprimir uma mensagem informando o número sorteado e se o jogador adivinhou ou não o segredo.

Para isso, deve-se implementar duas funções:

```
def posicoes_certas(s, c):  
    '''(int, int) -> int  
    Recebe dois números inteiros  $s \geq 0$  e  $c \geq 0$  com 3 dígitos  
    cada e retorna o número de posições certas de s e c.  
    Exemplos:  
    posicoes_certas(467,746) retorna 0  
    posicoes_certas(123,23) retorna 2  
    posicoes_certas(1,1) retorna 3  
    posicoes_certas(1,21) retorna 2  
    posicoes_certas(21,2) retorna 1  
    posicoes_certas(21,21) retorna 3  
    posicoes_certas(321,21) retorna 2  
    posicoes_certas(0,0) retorna 3  
    '''  
  
def digitos_casados(s, c):  
    '''(int, int) -> int  
    Recebe dois números inteiros  $s \geq 0$  e  $c \geq 0$  com 3 dígitos  
    cada e retorna o número dígitos casados entre s e c.  
    Exemplos:  
    digitos_casados(123,312) retorna 3  
    digitos_casados(123,31) retorna 2  
    digitos_casados(123,1) retorna 1
```



```
digitos_casados(100,1) retorna 3  
digitos_casados(100,102) retorna 2  
digitos_casados(2,1) retorna 2  
digitos_casados(0,0) retorna 3  
digitos_casados(222,1) retorna 0  
...
```




Gabarito

Exercícios de Fixação

1.

```
def main():
    user1 = input("Usuario 1, pedra, papel ou tesoura? ")
    user2 = input("Usuario 2, pedra, papel ou tesoura? ")

    if user1 == user2:
        print("Empate!")
    elif user1 == "pedra":
        if user2 == "papel":
            print("Usuario 2 vence!")
        else:
            print("Usuario 1 vence!")
    elif user1 == "papel":
        if user2 == "tesoura":
            print("Usuario 2 vence!")
        else:
            print("Usuario 1 vence!")
    elif user1 == "tesoura":
        if user2 == "pedra":
            print("Usuario 2 vence!")
        else:
            print("Usuario 1 vence!")
    else:
        print("Opa, algo deu errado, cheque suas entradas!")

main()
```

2.

```
def main():
    num = int(input("Digite um inteiro: "))
    soma = 0

    while num != 0:
        soma = soma + num
        num = int(input("Digite um inteiro: "))

    print("A soma e", soma)

main()
```



Exercícios de Prova

1. As saídas impressas pelo programa são: 28, 49, 14, 35 e 49.

2.

a. “A”, “E”, “F”, “H” e “I”.

b. “A”, “D”, “F”, “I” e “J”.

3.

```
import random

def posicoes_certas(s, c):
    certos = 0
    i = 0
    while i < 3:
        # Checa se o último dígito dos números são iguais
        if s % 10 == c % 10:
            certos += 1
        # tira o último dígito
        s = s // 10
        c = c // 10
        i = i + 1

    return certos

def digitos_casados(s, c):
    casamentos = 0
    s0 = s % 10
    s1 = (s // 10) % 10
    s2 = (s // 100)

    c0 = c % 10
    c1 = (c // 10) % 10
    c2 = c // 100

    # Compara o primeiro dígito de um número com todos do outro
    if s0 == c0:
        casamentos += 1
        c0 = -1 # garante que esse dígito não vai casar novamente
    elif s0 == c1:
        casamentos += 1
        c1 = -1
    elif s0 == c2:
        casamentos += 1
```



```
    c2 = -1
# Idem para o segundo dígito
if s1 == c0:
    casamentos += 1
    c0 = -1
elif s1 == c1:
    casamentos += 1
    c1 = -1
elif s1 == c2:
    casamentos += 1
    c2 = -1

# O último só pode casar com um dos outros
# Checa todos de uma vez
if s2 == c0 or s2 == c1 or s2 == c2:
    casamentos += 1

return casamentos

def main():
    # sorteio é um número aleatório entre 0 e 1000
    segredo = random.randrange(0,1000)

    certo = False
    tentativas = 0

    # Executa o programa enquanto número de tentativas for
    # menor que 6 e não tiver acertado
    while tentativas < 6 and not certo:
        tentativas += 1
        print("%da tentativa." % tentativas)
        chute = int(input("Chute: "))

        pos_certas = posicoes_certas(segredo, chute)
        dig_certos = digitos_casados(segredo, chute)

        print("Posicoes certas:", pos_certas)
        print("Digitos casados:", dig_certos)
        print()
        if segredo == chute:
            certo = True

    if certo:
        print("Você adivinhou o segredo", segredo, "em", tentativas, "tentativas")
    else:
        print("Você não adivinhou o segredo", segredo)

main()
```