Atividade de Recuperação – 1° VA

Anápolis, 29 de março de 2024.

Discentes: Matheus Marques Portela -2310823,

Nome da disciplina: Laboratório de Programação

Correção Prova

1. Assinale a afirmação verdadeira sobre funções em Python.

d. Uma função pode retornar múltiplos valores em uma única instrução return.
(Em Python, uma função pode retornar múltiplos valores em uma única instrução return usando uma tupla, lista, ou qualquer outra estrutura de dados que possa conter múltiplos valores.);

2. Sabemos que as funções, são estruturas essenciais de algoritmos, de acordo com o que foi visto em sala assinale a alternativa que descreve qual o conceito fundamental de uma função em algoritmos e que pode ser utilizado também na linguagem Python.

d. Uma função é um conjunto de instruções que realizam uma tarefa específica.
(No Python podemos usar para diversos fatores dentro do código, pois ele pode retornar um valor);

3. Considere a seguinte função recursiva: função recursiva (x: inteiro): inteiro início

```
def recursiva(x):
  if x == 1:
    return -x
  else:
    return -5 * recursiva(x - 1) + x
```

Qual é o valor retornado pela função se ela for chamada com x = 4?
e. 164

4. Suponha que você tenha uma função Python que recebe uma lista de números e retorna a média aritmética dos valores. Se a lista estiver vazia, a função retorna NONE. Assinale a alternativa que seria uma implementação correta para essa função.

c.
```
def calcula_media(lista):
    if len(lista) == 0:
        return None
    return sum(lista) / len(lista)
```

(Aqui para retornar como NONE precisa saber se a lista está vazia e para isso usamos a função **LEN()**);

5. Qual é o termo utilizado para descrever a passagem de argumentos para uma função, onde os valores dos argumentos são passados diretamente para os parâmetros?

d. Passagem por valor
(O termo utilizado para descrever a passagem de argumentos para uma função, onde os valores dos argumentos são passados diretamente para os parâmetros, é o valor);

6. Ao chamar uma função em Python, é esperado o comportamento padrão dos argumentos caso nenhum valor seja fornecido para eles, assinale a alternativa correta que descreve como os argumentos serão passados neste caso.

b. Os argumentos recebem valores padrão definidos pelo programador.
(Em Python, é possível definir valores padrão para os argumentos de uma função. Se nenhum valor for fornecido para um argumento durante a chamada da função, o valor padrão definido pelo programador será utilizado.);

7. (Quadrix - 2019 - Prefeitura de Jataí - GO - Analista de Tecnologia da Informação). A situação em que dois subprogramas fazem chamadas recíprocas, como, por exemplo, um subprograma P faz uma chamada a um subprograma J, que, por sua vez, faz uma chamada a P, é caracterizada como uma

b. Recursividade Indireta
(Na recursividade indireta, dois ou mais subprogramas estão interligados em um ciclo de chamadas.);

8. (PR-4 UFRJ - 2018 - UFRJ - Analista de Tecnologia da Informação). Assinale a alternativa que define corretamente a técnica de função fatorial empregada no pseudocódigo a seguir.

```
funcao fatorial(n)
se n=1 então
        fatorial = 1
senão
       fatorial = n * fatorial(n-1)
fim funcao
```

a. Recursividade
(A função que chama a si mesma denominação recursiva);

9. (CONSULPLAN - 2017 - TRE-RJ - Técnico Judiciário - Programação de Sistemas). Analise as afirmativas a seguir a respeito de algoritmos recursivos.

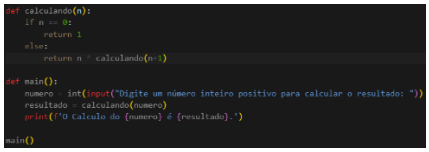
I. Diz-se que uma rotina é recursiva se a sua definição envolver uma chamada a ela mesma. Neste sentido, o termo recursão é equivalente ao termo indução utilizado por matemáticos.
II. Cada algoritmo recursivo possui um algoritmo iterativo equivalente e vice-versa, mas que pode ter mais ou menos complexidade em sua construção.
III. Uma função recursiva possui duas partes: caso base e caso recursivo.
IV. Um algoritmo pode ser chamado de iterativo quando ele requer a repetição implícita de um processo até que determinada condição seja satisfeita.
V. A recursividade possibilita a escrita de um código mais enxuto, com maior legibilidade e simplicidade.
Assinale a alternativa que possui alguma afirmação INCORRETA.

c. III e IV
(III. Uma função recursiva geralmente é composta por duas partes: o caso base, que é uma condição que indica quando a recursão deve parar, e o caso recursivo, que é a chamada recursiva em si.
IV. Algoritmos iterativos envolvem repetição explícita de um processo até que determinada condição seja satisfeita, enquanto algoritmos recursivos envolvem a chamada a si mesmos para resolver um problema de maneira mais elegante.

10. (Instituto Darwin - 2023 - Prefeitura de Lagoa de Itaenga - PE - Técnico em Informática). Quanto a Recursividade é INCORRETO afirmar que:

d. Um programa recursivo é mais elegante e menor que a sua versão iterativa, além de exibir com maior clareza o processo utilizado, desde que o problema ou os dados sejam naturalmente definidos através de recorrência. Além de exigir um menor espaço de memória e é, na grande maioria dos casos, mais rápido do que a versão iterativa.
(A recursão pode ser uma técnica elegante para resolver certos tipos de problemas, mas nem sempre resulta em um código mais curto ou mais rápido do que a versão iterativa.)

11. Se inserirmos um valor "-5" na variável "numero" da função abaixo, qual resultado será mostrado pelo programa principal?



```
def calculando(n):
    if n == 0:
        return 1
    else:
        return n * calculando(n-1)

def main():
    numero = int(input("Digite um número inteiro positivo para calcular o resultado: "))
    resultado = calculando(numero)
    print(f'O Calculo do {numero} é {resultado}.')

main()
```

a. -120

12. (UFSC - 2019 - UFSC - Técnico de Tecnologia da Informação). A respeito de um algoritmo recursivo, analise as afirmativas abaixo e assinale a alternativa correta.

I. Deve conter pelo menos uma estrutura de repetição.

II. Deve conter pelo menos uma estrutura de seleção.

III. Deve invocar a si mesmo pelo menos uma vez ao ser executado.

c. Somente as alternativas II e III estão corretas
(II. Um algoritmo recursivo geralmente contém pelo menos uma estrutura de seleção para definir um caso base, que é a condição de parada da recursão.
III. A definição de um algoritmo recursivo é que ele chama a si mesmo para resolver subproblemas menores do mesmo tipo até atingir um caso base.)

13. Sabemos que para declarar uma função devemos, inserir a palavra chave que define a função, o nome da função e a lista de parâmetros que podem ser necessários ou não. Analise o código a seguir e assinale a alternativa correta. "def Soma(int a, int b)":

e. def Soma(a, b):

14. (FCC - 2015 - DPE-SP - Programador). O uso da recursividade geralmente permite uma descrição mais clara e concisa dos algoritmos. Em relação aos conceitos e utilização de recursividade, é correto afirmar:

b. Na prática, é necessário garantir que o nível mais profundo de recursão seja finito e que também possa ser mantido pequeno, pois em cada ativação recursiva de um procedimento P, uma parcela de memória é requerida.
(É importante controlar o número de chamadas recursivas e o tamanho da pilha de chamadas para evitar estouro de memória.)

15. Considere a seguinte função Python:

```
def g(n):
    return n * g(n - 1)
```

Chamando g(4), qual será o resultado retornado?

b.NDA

16. Considere a seguinte função Python:

```
def f(x):
    if x == 0:
        return 1
    return x * f(x - 1)
```

Qual será o resultado mostrado se f(4) é chamado?

e.24

17. Considera a função recursiva a seguir:

```
def argu(num):
  if num == 0:
    return 0
  else:
    return (num % 2) + 10 * argu(num//2)

print(argu(5))
```

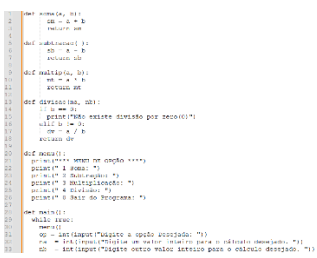
Qual o valor retornado pela função acima, quando recebe como parâmetro o número 5?

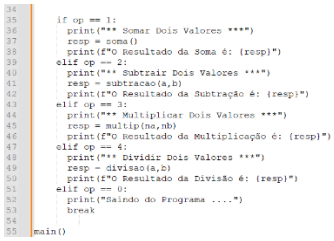
d.101

18. (SUGEP - UFRPE - 2018 - UFRPE - Técnico de Tecnologia da Informação - Sistemas). Considere a função recursiva ‘func’ definida por
func(1) = 1
func(n) = (n - 1) * func(n - 1)

d.6 e 24

19. Observe o Código escrito em linguagem Python abaixo:





```
    if op == 1:
      print("** Somar Dois Valores ****")
      resp = soma()
      print(f"O Resultado da Soma é: {resp}")
    elif op == 2:
      print("** Subtrair Dois Valores ****")
      resp = subtracao(a,b)
      print(f"O Resultado da Subtração é: {resp}")
    elif op == 3:
      print("** Multiplicar Dois Valores ****")
      resp = multip(na,nb)
      print(f"O Resultado da Multiplicação é: {resp}")
    elif op == 4:
      print("** Dividir Dois Valores ****")
      resp = divisao(a,b)
      print(f"O Resultado da Divisão é: {resp}")
    elif op == 0:
      print("Saindo do Programa ....")
      break

main()
```

Faça uma análise do código, pontue exatamente, se houver, quais e onde estão os erros presentes no código e, reescreva todo o código sem os erros apontados. Considere, códigos sem erros, o algoritmo executando sem ter erros de sintaxe ou de lógica.

```
def soma(a,b):
    sm = a + b
    return sm

def subtracao(a, b):
    sb = a - b
    return sb

def multip(a,b):
    mt = a * b
    return mt

def divisao(a, b):
    if b == 0:
        return print("Não existe divisão por zero(0) ")
    else:
        dv = a / b
        return dv

def menu():
    print(" 1 soma:  ")
    print(" 2 Subtração:  ")
    print(" 3 Multiplicação:  ")
    print(" 4 Divisão:  ")
    print(" 0 Sair do programa:  ")

def main():
    while True:
        menu()
        op = int(input("Digite a opção desejada: ")
        na = int(input("Digite um valor inteiro para o calculo desejado: ")
        nb = int(input("Digite outro valor inteiro para o calculo desejado: ")

        if op == 1:
            print("**  Somar dois valores *** ")
            resp = soma(na, nb)
            print(f"O resultado da soma é: {resp}")
        elif op == 2:
            print("**  Subtrair dois valores *** ")
            resp = subtracao(na, nb)
            print(f"O resultado da subtração é: {resp}")
        elif op == 3:
            print("**  Multiplicar dois valores *** ")
            resp = multip(na, nb)
            print(f"O resultado da multiplicação é: {resp}")
        elif op == 4:
```

```
            print("**  Dividir dois valores *** ")
            resp = divisao(na, nb)
            print(f"O resultado da divisao é: {resp}")
        elif op == 0:
            print("  Saindo do programa... ")
            break
main()
```

20. (AOCP - 2018 - FUNPAPA - Analista de Sistemas). Existem casos em que um procedimento ou função chama a si próprio. Sobre introdução à computação, é correto afirmar que:

d. quando um procedimento ou função chama a si próprio, denomina-se recursividade.

21. Escreva um código de exemplo em Python para cada um dos tipos de funções a seguir:

Função sem retorno que recebe três parâmetros

Função com retorno e que não recebe parâmetros

Função sem retorno e que não recebe parâmetros

Função com retorno que recebe dois parâmetros

Função que chama todas as outras funções acima

```
def func1(a, b, c):         #Função sem retorno mas recebe parâmetros
    soma = a + b + c
    print(f"A soma dos valores é {soma}")

def func2():         #Função com retorno e não recebe parâmetros
    n1 = int(input('Digite um número: ')
    n2 = int(input('Digite outro número: ')
    soma = n1 + n2
    return soma

def func3():         #Função sem retorno e não recebe parâmetros
    n1 = int(input('Digite um número: ')
    n2 = int(input('Digite outro número: ')
    soma = n1 + n2
    print('A soma dos valores é {soma}')

def func4(a, b):         #Função com retorno e recebe parâmetros
    soma = a + b
    return soma

def main():
   n1 = int(input('Digite um número: ')
   n2 = int(input('Digite outro número: ')
   n3 = int(input('Digite outro número: ')
```

```
    func1(n1, n2, n3)

    result = func2()
    print(result)

    func3()

    result2 = func4(n1, n2)
    print(result2)

main()
```

22. Observe o código abaixo:

```python
def calculando(n):
    if n == 0:
        return 1
    else:
        return n * calculando(n-1)

def main():
    numero = int(input("Digite um número inteiro positivo para calcular o resultado: "))
    resultado = calculando(numero)
    print(f'O Calculo do {numero} é {resultado}.')

main()
```

assinale a alternativa que será exibido na tela, como resultado, se a variável "numero" receber o valor 6.

c.120