



## Questão 9

Correto

Atingiu 1,00 de 1,00

Padrão triangular?

O [programa](#) a seguir deveria imprimir 8 linhas exibindo um padrão visualmente triangular usando-se apenas as [instruções](#) e outros recursos que aprendemos até agora. Corrija os erros de [sintaxe](#) e de lógica para que ele imprima o padrão 16 vezes como em uma tabela 4x4.

```
s1, s2, s3 = input().split()

def duplica(f):
    f()
    f()
    f()

def duplica_o_duplicado(f):
    duplica(f)
    duplica(f)

def imprime_parte_linha1():
    print(" "+s1+s2+" "+, end="")

def imprime_parte_linha1_fim():
    print("")

def imprime_parte_linha2():
    print(s1+2*s3+s2, end="")

def imprime_parte_linha2_fim():
    print(" ")

def imprime_linha1():
    duplica_o_duplicado(imprime_parte_linha1)
    imprime_parte_linha1_fim()

def imprime_linha2():
    duplica_o_duplicado(imprime_parte_linha2)
    imprime_parte_linha2_fim()

def imprime_parte_padrao():
    imprime_linha1()
    imprime_linha2()

def imprime_padrao():
    duplica_o_duplicado(imprime_parte_padrao)

imprime_padrao()
```

### A Entrada consiste de:

- 3 caracteres em uma linha, separados por espaços em branco. Cada caracter representa um lado do triângulo.

### A Saída deve apresentar:

- O resultado deve ser um padrão triangular 4x4

### Observações:

- Não é necessário validar se os valores de entrada estão dentro dos intervalos definidos.
- Relembre as [funções](#) apresentadas no Capítulo 3.



Descrição dos Exemplos:

- Para a entrada "/ \ \_" a saída deve ter o seguinte aspecto visual:

```

  /\  /\  /\  /\
/_\/_\/_\/_\
  /\  /\  /\  /\
/_\/_\/_\/_\
  /\  /\  /\  /\
/_\/_\/_\/_\
  /\  /\  /\  /\
/_\/_\/_\/_\

```

For example:

Input	Result
/ \ _	/\  /\  /\  /\           /\/_\/_\/_\/_\           /\  /\  /\  /\           /\/_\/_\/_\/_\           /\  /\  /\  /\           /\/_\/_\/_\/_\           /\  /\  /\  /\           /\/_\/_\/_\/_\
^ \ -	^\  ^\  ^\  ^\           ^--\^--\^--\^--\           ^\  ^\  ^\  ^\           ^--\^--\^--\^--\           ^\  ^\  ^\  ^\           ^--\^--\^--\^--\           ^\  ^\  ^\  ^\           ^--\^--\^--\^--\
\ / -	\/\  \/\  \/\  \/\           \--\/\--\/\--\/\--/           \/\  \/\  \/\  \/\           \--\/\--\/\--\/\--/           \/\  \/\  \/\  \/\           \--\/\--\/\--\/\--/           \/\  \/\  \/\  \/\           \--\/\--\/\--\/\--/

Answer: (penalty regime: 0, 0, 10, 20, ... %)

RESET ANSWER

```

1  #Realize as alterações necessárias no código apresentado.
2  s1, s2, s3 = input().split()
3
4  def duplica(f):
5      return f+f
6  def duplica_o_duplicado(f):
7      print(duplica(f)+duplica(f))
8
9  def imprime_parte_linha1(s1,s2):
10     return (" "+s1+s2+" ")
11
12  def imprime_parte_linha1_fim():
13     return "\n"
14
15  def imprime_parte_linha2(s1,s2,s3):
16     return s1+2*s3+s2
17
18  def imprime_parte_linha2_fim():
19     return " \n"
20

```



```

21 ▾ def imprime_linha1(s1,s2,s3):
22     duplica_o_duplicado((imprime_parte_linha1(s1,s2)))
23     duplica_o_duplicado((imprime_parte_linha2(s1,s2,s3)))
24
25 ▾ def imprime_linha2(s1,s2,s3):
26     duplica_o_duplicado((imprime_parte_linha1(s1,s2)))
27     duplica_o_duplicado((imprime_parte_linha2(s1,s2,s3)))
28
29 ▾ def imprime_parte_padrao(s1,s2,s3):
30     imprime_linha1(s1,s2,s3)
31     imprime_linha2(s1,s2,s3)
32
33 ▾ def imprime_padrao(s1,s2,s3):
34     imprime_parte_padrao(s1,s2,s3)
35     imprime_parte_padrao(s1,s2,s3)
36
37 imprime_padrao(s1,s2,s3)
38

```

PRECHECK VERIFICAR

	Input	Expected	Got	
✓	/ \ _	/\ /\ /\ /\           /_\/_\/_\/_\/           /\ /\ /\ /\           /_\/_\/_\/_\/           /\ /\ /\ /\           /_\/_\/_\/_\/           /\ /\ /\ /\           /_\/_\/_\/_\/	/\ /\ /\ /\           /_\/_\/_\/_\/           /\ /\ /\ /\           /_\/_\/_\/_\/           /\ /\ /\ /\           /_\/_\/_\/_\/           /\ /\ /\ /\           /_\/_\/_\/_\/	✓
✓	^ \ -	^\ ^\ ^\ ^\           ^--^--^--^--\           ^\ ^\ ^\ ^\           ^--^--^--^--\           ^\ ^\ ^\ ^\           ^--^--^--^--\           ^\ ^\ ^\ ^\           ^--^--^--^--\	^\ ^\ ^\ ^\           ^--^--^--^--\           ^\ ^\ ^\ ^\           ^--^--^--^--\           ^\ ^\ ^\ ^\           ^--^--^--^--\           ^\ ^\ ^\ ^\           ^--^--^--^--\	✓
✓	\ / -	\/\ \/\ \/\ \/\           \--/\--/\--/\--/           \/\ \/\ \/\ \/\           \--/\--/\--/\--/           \/\ \/\ \/\ \/\           \--/\--/\--/\--/           \/\ \/\ \/\ \/\           \--/\--/\--/\--/	\/\ \/\ \/\ \/\           \--/\--/\--/\--/           \/\ \/\ \/\ \/\           \--/\--/\--/\--/           \/\ \/\ \/\ \/\           \--/\--/\--/\--/           \/\ \/\ \/\ \/\           \--/\--/\--/\--/	✓
✓	* * *	** ** ** **           *****           ** ** **           *****           ** ** **           *****           ** ** **           *****	** ** **           *****           ** ** **           *****           ** ** **           *****           ** ** **           *****	✓
✓	[ ] _	[] [] [] []           [] [] [] []           [] [] [] []           [] [] [] []           [] [] [] []           [] [] [] []           [] [] [] []	[] [] [] []           [] [] [] []           [] [] [] []           [] [] [] []           [] [] [] []           [] [] [] []           [] [] [] []	✓



	Input	Expected	Got	
✓	. . .	. . . . . ..... . . . . . ..... . . . . . ..... . . . . . .....	. . . . . ..... . . . . . ..... . . . . . ..... . . . . . .....	✓

Passou em todos os teste! ✓

Correto

Notas para este envio: 1,00/1,00.

