- **Amirhossein rasouli**
- **HW-Week3**

1. In Listing 5.4 (addnonnegatives.py) could the condition of the

if statement have used > instead of >= and achieved the same

results? Why?

yes, we could use if entry > -1, this works because just like entry >= 0,

this too includes 0 and excludes all negative number..

---

2. In Listing 5.4 (addnonnegatives.py) could the condition of the

while statement have used > instead of >= and achieved the

same results? Why?

yes, like said above we could use while entry > -1, and it would work

as intended.

---

3. In Listing 5.4 (addnonnegatives.py) what would happen if the

statement

entry = int(input()) # Get the value

were moved out of the loop? Is moving the assignment out of

the loop a good or bad thing to do? Why?

if we write this line above the loop, then according to the entry theuser gives, we could have either a infinite loop (if entry >= 0) or we could

have no loop at all (entry < 0).

but if moved under the loop, we wont take any input at all because the

while loop would work infinitly.

---

4. How many asterisks does the following code fragment print?

```
a = 0
while a < 100:
    print('*', end='')
    a += 1
print()
```

100 asterisks.

---

5. How many asterisks does the following code fragment print?

```
a = 0
while a < 100:
    print('*', end='')
print()
```

infinite asterisks.

---

6. How many asterisks does the following code fragment print?

```
a = 0
while a > 100:
    print('*', end='')
    a += 1
print()
```

no asterisks.

---

7. How many asterisks does the following code fragment print?

```
a = 0
while a < 100:
    b = 0
    while b < 55:
        print('*', end='')
        b += 1
    print()
    a += 1
```

100 lines, each line 55 asterisks.

---

8. How many asterisks does the following code fragment print?

```
a = 0
while a < 100:
    if a  % 5 == 0:
        print('*', end='')
    a += 1
print()
```

20 asterisks.

---

9. How many asterisks does the following code fragment print?

```
a = 0
while a < 100:
    b = 0
    while b < 40:
        if (a + b) % 2 == 0:
            print('*', end='')
        b += 1
    print()
    a += 1
```

adding two odd values gives us a even one,

and adding two even values, will always give us a even value.

zin 40 we have 20 odd values, and 20 even values.

so for each line, if we get a odd value for a, 20 odd b values will be

added to it and make it even,

and if we get a even value for a, 20 even b values will be added to it and

make it even again.

in conclusion:

100 lines, each line 20 asterisks.

---

10. How many asterisks does the following code fragment print?

```
a = 0
while a < 100:
    b = 0
    while b < 100:
        c = 0
        while c < 100:
            print('*', end='')
            c += 1
        b += 1
    a += 1
print()
```

100 to the power of 3 so 1,000,000.

---

11. What is minimum number of arguments acceptable to the range expression?

One

---

12. What is maximum number of arguments acceptable to the range expression?

Three

---

13. Provide the exact sequence of integers specified by each ofthe following range expressions.

(a) range(5)

0, 1, 2, 3, 4(b) range(5, 10)

5, 6, 7, 8, 9

_____

(c) range(5, 20, 3)

5 ,8 ,11 ,14 ,17

_____

(d) range(20, 5, -1)

20 ,19 ,18 ,17 ,16 ,15 ,14 ,13 ,12 ,11 ,10 ,9 ,8 ,7 ,6

_____

(e) range(20, 5, -3)

20 ,17 ,14 ,11 ,8

_____

(f) range(10, 5)

nothing, since we have to add -1 if we are going back wards.

_____

(g) range(0)

nothing.

_____

(h) range(10, 101, 10)

10 ,20 ,30 ,40 ,50 ,60 ,70 ,80 ,90 ,100

_____

(i) range(10, -1, -1)

10 ,9 ,8 ,7 ,6 ,5 ,4 ,3 ,2 ,1 ,0

_____

(j) range(-3, 4)

-3 ,-2 ,-1 ,0 ,1 ,2 ,3

_____

(k) range(0, 10, 1)

0 ,1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9

__-_____

14. What is a shorter way to express range(0, 5, 1)?

range(5)

__-_____

15. Provide an equivalent Python range expression for each of the following integer sequences.

(a) 1,2,3,4,5

range(1, 6)

_____

(b) 5,4,3,2,1

range(5, 0, -1)

_____

(c) 5,10,15,20,25,30

range(5, 31, 5)

_____

(d) 30,25,20,15,10,5

range(30, 4, -5)

_____

(e) -3,-2,-1,0,1,2,3

range(-3, 4)

_____

(f) 3,2,1,0,-1,-2,-3

range(3, -4, -1)

_____

(g) -50,-40,-30,-20,-10

range(-50, -9, 10)

_____

(h) Empty sequence

range(0)

__-_____

16. If x is bound to the integer value 2, what integer sequence does range(x, 10*x, x) represent?

range (2 , 20 , 2) ---> 2 , 4 , 6 , ... ,
18

__-_____

17. If x is bound to the integer value 2 and y is bound to the integer 5, what integer sequence does
range(x, x + y)
represent?

range (2 , 7) ---> 2 , 3 , 4 , 5 ,
6

18. Is it possible to represent the following sequence with a Python range expression: 1;-1;2;-2;3;-3;4;-4 ?

Who knows ? :)

---

19. How many asterisks does the following code fragment print? ---> 99

```
for a in
range(100):
 print('*',
end='')

print()
```

---

20. How many asterisks does the following code fragment print? ---> 16 / 20 , 25 , ... , 95

```
for a in range(20, 100,
5):
 print('*',
end='')
 print()
```

---

21. How many asterisks does the following code fragment print? ---> 50 / 100 - 98 - ... - 2

```
for a in range(100, 0, -
2):
 print('*',
end='')
 print()
```

---

22. How many asterisks does the following code fragment print? ---> Empty sequence

```
for a in range(1,
1):
 print('*',
end='')
 print()
```

23. How many asterisks does the following code fragment print? ---> 200 / -100 - -99 - ... - 99

```
for a in range(-100,
100):
 print('*',
end='')

print()
```

---

24. How many asterisks does the following code fragment print? ---> 20 / -100 - -90 - ... - 90

```
for a in range(-100, 100,
10):
 print('*',
end='')

print()
```

---

25. Rewrite the code in the previous question so it uses a while instead of a for. Your code should
behave
identically.

```
counter =
0
number = -
100
while number <
100 :

print("*",end="")
 number +=
10
 counter +=
1
 print("\nCounter = ",counter) # For count number of
asterisks ;)
```

---

26. What does the following code fragment
print?

```
a =
```

```
0
 while a <
100:

print(a)
 a +=
1

print()
```

output --->

```
0

1

2
 ..
.

99
```

---

27. Rewrite the code in the previous question so it uses a for instead of a while. Your code should behave identically.

```
 for number in range (1 ,
100):
 print(number)
```

---

28. What is printed by the following code fragment? ---> Empty sequence

```
 a =
0
 while a >
100:

print(a)
 a +=
```

1

print()

---

29. Rewrite the following code fragment using a break statement and eliminating the done variable.
Your code should behave identically to this code
fragment.

```
done =
False
n, m = 0,
100
while not done and n !=
m:
n =
int(input())
if n <
0:
done =
True
print("n =",
n)
```

Rewrite --
->

```
n =
0
while n >=
0 :
n =
int(input())
if n <
0:

break
print("n =", n)
```

---

30. Rewrite the following code fragment so it eliminates the continue statement. Your new code’s logic

should be simpler than the logic of this
fragment.

```
x =
5
while x >
0:
y =
int(input())
if y ==
25:

continue
x -=
1
print('x =',
x)
```

Rewrite --
->

```
for loop in range
(5):
y =
int(input())
if y !=
25 :
print('x =', loop +
1)
```

---

31. What is printed by the following code fragment? ---> 0 1 2 ...
99

```
a =
0
while a <
100:
print(a, end='
')
a +=
1

print()
```

---

32. Write a Python program that accepts a single integer value entered by the user. If the value entered is
less than one, the program prints nothing. If the user enters a positive integer, n, the program prints
an n * n box drawn with * characters. If the users enters 1, for example, the program prints

```
- *
-
```

==========

```
<!- THE CODE
-!>


number_of_asterisks = int(float(input("How many asterisks do want to see Ma men ?
")))
if number_of_asterisks >=
1 :
asterisks =
number_of_asterisks
while number_of_asterisks !=
0 :
char = '*'*
asterisks

print(char)
number_of_asterisks -=
1
```

==========

---

33. Write a Python program that allows the user to enter exactly twenty floating-point values. The program
then prints the sum, average (arithmetic mean), maximum, and minimum of the values entered.

```
============================


 <!- THE CODE
-!>


 number_of_value =
20


print("\
n\t\t\tGeektory
.fun :)\n\n")
 print("\t\tGithub :
https://github.com/alunkom")
 print("\n","----"*6 ,
"\n")

 adad = float(input("Enter adad =
"))
 min , max =
adad ,adad
 sum , avrage = adad ,
1

 while number_of_value !=
1 :
 value = float(input("Enter adad =
"))
 sum +=
value
 avrage +=
1
 if value >
max :
 max =
value
 elif value <
min :
 min =
value
 number_of_value -=
1

 print("\n","----"*6 ,
"\n")
 print("Min of numbers
```

```
= %f"%min)
 print("Max of numbers
= %f"%max)
 print("Avrage if numbers
= %f"%(sum/avrage))
 print("Sum of numbers = %f"%sum)
```

---

34. Write a Python program that allows the user to enter any number of nonnegative floating-point values.
The user terminates the input list with any negative value. The program then prints the sum, average
(arithmetic mean), maximum, and minimum of the values entered. The terminating negative value is
not used in the
computations.

==========

```
 <!- THE CODE
-!>



print("\
n\t\t\tGeektory
.fun :)\n\n")
 print("\t\tGithub :
https://github.com/alunkom")
 print("\n","----"*6 ,
"\n")

 adad = float(input("Enter posetive adad [negetive for end] =
"))

 if adad <
0 :
 print("\n\t\t\tBye Bye ma raftim :)
")

else
:
```

```
 min , max =
adad ,adad
 sum , avrage = adad ,
1
 condition =
True

 while
condition :
 value = float(input("Enter adad =
"))
 if value <
0 :
 print("\n\t\t\tBye Bye ma raftim :)
")
 condition =
False

else:
 sum +=
value
 avrage +=
1
 if value >
max :
 max =
value
 elif value <
min :
 min =
value

 print("\n","----"*6 ,
"\n")
 print("Min of numbers
= %f"%min)
 print("Max of numbers
= %f"%max)
 print("Avrage if numbers
= %f"%(sum/avrage))
 print("Sum of numbers
= %f"%sum)
```

---

35. Redesign Listing 5.34 (startree.py) so that it draws a sideways tree pointing right; for example, if the
user enters 7, the program would

print

==========

<!- THE CODE
-!>

adad = int(input("Enter number :
"))

row_bala = adad -
1

for row in range(1 ,
row_bala+1):

print("*"*row)

print("*"*adad)

for row_paiin in range (row_bala , 0 , -
1):

print("*"*row_paiin)

---

36. Redesign Listing 5.34 (startree.py) so that it draws a sideways tree pointing left; for example, if the
user enters 7, the program would
print :

==========

<!- THE CODE
-!>

```
print("\
n\t\t\tGeektory
.fun :)\n\n")
 print("\t\tGithub :
https://github.com/alunkom")
 print("\n","----"*6 ,
"\n")

 # Declare variables and input
section

 adad = int(input("Enter number :
"))
 row_space = adad -
1
 number_of_asterisks =
1
 asterisks = '*' *
number_of_asterisx


 for space__bala in range(row_space , 0 , -
1):
 print(" "*space__bala ,
asterisks,sep='')
 number_of_asterisks +=
1
 asterisks = number_of_asterisks *
'*'


print("*"*adad)

 for space_paiin in range(1 ,
row_space+1 ):
 number_of_asterisks -=
1
 asterisks = '*' *
number_of_asterisks
 print(' '*space_paiin,asterisks,sep='')
```