Week-3 Practice Assignment (Theory)

Week-3 Practice Assignment (Theory)

- Problem 1
  - Question
  - Answer
  - Solution
  - Tags
- Problem 2
  - Question
  - Answer
  - Solution
  - Tags
- Problem 3
  - Question
  - Answer
  - Solution
  - Tags
- Problem 4
  - Question
  - Answer
  - Solution
  - Tags
- Problem 5
  - Question
  - Answer
  - Solution
  - Tags
- Problem 6
  - Question
  - Answer
  - Solution
  - Tags
- Problem 7
  - Question
  - Answer
  - Solution
  - Tags
- Problem 8
  - Question
  - Answer
  - Solution
- Problem 9
  - Question a
  - Answer
  - Solution
  - Question b
  - Answer
  - Solution

Problem 1

Question

`x` is a variable of type `float` that has already been defined. Which of the following options could be the output of the following code-block? It is a Multiple Select Question (MSQ).

```
print(f'{x:1.2f}')
```

(a) `1.2`

(b) `1.29`

(c) `1234.20`

(d) `1234.234`

Answer

(b), (c)

Solution

```
print(f'{x:1.2f}')
```

The specification is that there must be two places after the decimal point.

- If the variable `x` has more than two places after the decimal point, then it will be rounded off to exactly two places.
- If the variable `x` has less than two places after the decimal point, then zeros will be appended at the end so that there are exactly two places after the decimal point.

Only options (b) and (c) conform to this specification.

Tags

basicio

Problem 2

Question

What does the following code-block print?

```
n = int(input("Enter any positive number: "))
for i in range(1,11):
    print(n * i)
```

(a) First 11 multiples of `i`.

(b) First 10 multiples of `i`.

(c) First 10 multiples of `n`.

(d) First 11 multiples of `n`.

Answer

(c).

Solution

`n` stores the input number. `range(1, 11)` generates numbers from 1 to 10 and assigns each number to i in every iteration. In every iteration the value of (n * i) gets printed.

Tags

for loop.

Problem 3

Question

If the while loop of below code executes, when does it terminate?

```
x = int(input("Enter any number: "))
while(x % 5 != 0 and x % 10 != 0):
    x = int(input("Enter any number: "))
print("outside loop, the value of x is ", x)
```

(a) Never terminates it is a infinite loop.

(b) when we input a number which is multiple of 10.

(c) when we input a number which is multiple of 5.

(d) when we input a number which is not a multiple of both 5 and 10.

Answer

(b), (c)

Solution

while loop terminates if the condition `x % 5 != 0 and x % 10 != 0` becomes `False` it happens only when we input a number which is multiple of 5. (multiple of 10 is always a multiple of 5).

Tags

while loop.

Problem 4

Question

What does `average` represent at the end of execution?

```
total = 0
count = 0
for i in range(1000):
    if i % 2 != 0 and count <= 50:
        total = total + i
        count = count + 1
average = total / count
print(average)
```

(a) average of 1 to 1000 numbers

(b) average of first 1000 odd numbers

(c) average of first 50 odd numbers

(d) average of first 51 odd numbers

Answer

(d)

Solution

There are 1000 iterations in the for-loop. The loop increments `total` and `count` by `i` and `1` respectively in each iteration when `i` is an odd number and `count` is less than or equal to 50.

Tags

for loop

Problem 5

Question

What does the printed value represents at the end of execution?

```
n = int(input("Enter any positive number: "))
i = 1
while (i <= n):
    if (n % i == 0):
        print(i)
    i = i + 1
```

(a) Multiples of `i`

(b) Factors of `i`

(c) Factors of `n`

(d) Multiples of `n`

Answer

(c) Factors of `n`

Solution

If a number `n` is divisible by the number `i`, then `i` is called a factor of `n`. When `i` is a factor of `n`, then `n % i == 0`.

Tags

Problem 6

Question

What is the output of the following code

```
total = 0
for i in range(1, 5):
    for j in range(i):
        total = total + i
print(total)
```

(a) 4

(b) 30

(c) 10

(d) 20

Answer

(b)

Solution

- In the first iteration of outer loop i = 1, the inner loop iterates 1 time and the total is 1
- In the second iteration of outer loop i = 2, the inner loop iterates 2 times and the total is 5
- In the third iteration of outer loop i = 3, the inner loop iterates 3 times and the total is 14
- In the fourth iteration of outer loop i = 4, the inner loop iterates 4 times and the total is 30

Tags

nested for loop

Problem 7

Question

How many asterisks does the following code print

```
i = -3
while (i <= 13):
    print('**')
    print('*')
    i = i + 1
```

(a) 51

(b) 25

(c) 39

(d) 48

Answer

(a)

Solution

In every iteration 3 asterisks will be printed, 17 iterations happens, hence 17 * 3 = 51 asterisks will be printed.

Tags

while loop.

Problem 8

Question

Which is the correct option that can be used in order to add ‘n-1’ blank spaces after a given string T?

Note: Variable T is a single character string.

(a) `print("%ns"%T)`

(b) `print("-ns"%T)`

(c) `print("%-ns"%T)`

(d) `print("-ns"%T)`

Answer

(c) `print("%-ns"%T)`

Solution

For any string T , `print("%ns"%T)` format method adds `(n-len(T))` blank space to the right of T if n is a negative integer and adds `n-len(T)` blank space to the left of T if n is a positive integer. `n-len(T)` should be greater than 0.

According to the problem statement length of T is 1 so, option (c) is correct to add `n-1` blank spaces after a given string T.

Problem 9

Question a

You are an analyst in a Finance company. You are given a job to print the daily transaction in below format.

```
country_code, currency_code, exchange_rate
```

Select the options that prints transaction records as in the sample output for input given in the table below. Multiple options can be correct (MSQ).

Input

Variable	VALUE
country_code	IN
currency_code	RS
exchange_rate	73.2272

Output

```
IN, RS, 73.23
```

(a) `print(country_code, currency_code, exchange_rate, sep = ", ")`

(b) `print(f"{country_code}, {currency_code}, {exchange_rate:2.2f}")`

(c) `print("{}, {}, {:2.2f}".format(country_code, currency_code, exchange_rate))`

(d) `print("{0}, {1}, {2:2.2f}".format(country_code, currency_code, exchange_rate))`

(e) `print("{a}, {b}, {c:2.2f}".format(a = country_code, b = currency_code, c = exchange_rate))`

(f) `print("%s, %s, %2.2f"%(country_code, currency_code, exchange_rate))`

Answer

(b), (c), (d), (e), (f)

Solution

Since, there is no format string used to round the float value `73.2272` to `2` decimal places, option (a) is incorrect. Option (b), (c), (d), (e), (f) are correct. These options use format string `.2f` to approximate `exchange_rate` to `2` decimal places.

Question b

In continuation to the last question you are also asked to capture the date and time of transaction in the below format.

```
day-month-year hour:minute:second
```

Select the options that prints transaction records as in the sample output for input given in the table below. Multiple options can be correct (MSQ).

Input

variable	VALUE
day	01
month	01
year	2021
hour	12
minute	30
second	00

Output

```
01-01-2021 12:30:00
```

(a)

```
print(day, month, year, sep = "-", end = " ")
print(hour, minute, second, sep = ":")
```

(b) `print(f"{day}-{month}-{year} {hour}:{minute}:{second}")`

(c) `print("{}-{}-{} {}:{}:{}".format(day, month, year, hour, minute, second))`

(d) `print("{0}-{1}-{2} {3}:{4}:{5}".format(day, month, year, hour, minute, second))`

(e) `print("{a}-{b}-{c} {d}:{e}:{f}".format(a = day, b = month, c = year, d = hour, e = minute, f = second))`

(f) `print("%s-%s-%s %s:%s:%s"%(day, month, year, hour, minute, second))`

Answer

(a), (b), (c), (d), (e), (f)

Solution

All options are valid and prints the expected output.