PYTHON WORK SHEET SET 03 Assignment 10

Q1 to Q8 have only one correct answer. Choose the correct option to answer your question.

1. Which of the following will raise a value error in python?

A) int(32) B) int(3.2) C) int(-3.2) D) int('32')

Answer: D) int('32')

2. What will be the output of round(3.567)?

A) 3.5 B) 3.0 C) 4 D) 3

Answer: C) 4

3. How is the function pow(a,b,c) evaluated in python?

A) a**b**c B) (a**b)%c C) (a**b)*c D) (a**b)**c

Answer: B) (a**b)%c

4. What will be the output of print(type(type(int))) in python 3?

A)<class 'type'> B)<type 'type'> C)<class 'int'> D)<type 'int'>

Answer: B)<type 'type'>

5. What will be the output of ord(chr(65))?

A) 'A' B) 'a' C) 65 D) TypeError

Answer: C) 65

6. What is called when a function is defined inside a class?

A) Module B) Function C) _init_ function D) Method

Answer: D) Method

7. What will be the output of all([1, 0, 5 ,7])?

A) 0 B) False C) True D) error

Answer: B) False

8. Is the output of the function abs() the same as that of the function math.fabs()?

A) Always B) Sometimes C) Never D) None of these

Answer: B) Sometimes

Q9 and Q10 have multiple correct answers. Choose all the correct options to answer your question.

9. Select all correct float numbers in python?

A) -68.7e100 B) 42e3 C) 4.2038 D) 3.0

Answer: B) 42e3, C) 4.2038, D) 3.0

10. Which of the following is(are) correct statement(s) in python?

A) You can pass positional arguments in any order.

B) You can pass keyword arguments in any order.

C) You can call a function with positional and keyword arguments.

D) Positional arguments must be before keyword arguments in a function call

Answer: A) You can pass positional arguments in any order.

B) You can pass keyword arguments in any order. C) You can call a function with positional and keyword arguments

Q11 to Q15 are programming questions. Answer them in Jupyter Notebook.

11. Write a python function print pyramid of stars. Level of the pyramid should be taken as an input from the user. E.g. Input = 5 Output

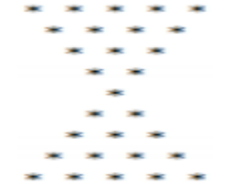
Answer:

```
# Define a function to print a pyramid of stars
def print_pyramid(level):
  # Loop from 1 to level
  for i in range(1, level + 1):
    # Print spaces before stars
    for j in range(level - i):
      print(" ", end="")
    # Print stars
    for k in range(2 * i - 1):
      print("*", end="")
    # Print newline after each row
    print()

# Take user input for level
try:
  level = int(input("Enter the level of the pyramid: "))
except ValueError:
  # Invalid input type
  print("Please enter a valid integer")
else:
  # Check if level is positive
  if level > 0:
    # Call the function with user input value
    print_pyramid(level)
  else:
    # Invalid input for pyramid level
    print("Please enter a positive integer")
```

```
Enter the level of the pyramid: 5
    *
   ***
  *****
 *******
*********
```

12. Write a python function print Hourglass pattern. E.g. Input = 5 Output:



Answer:

```
# Define a function to print an hourglass pattern
def print_hourglass(rows):
  # Loop from rows to 1
  for i in range(rows, 0, -1):
    # Print spaces before stars
    for j in range(rows - i):
      print(" ", end="")
    # Print stars
    for k in range(2 * i - 1):
      print("*", end="")
    # Print newline after each row
    print()
  # Loop from 2 to rows
  for i in range(2, rows + 1):
    # Print spaces before stars
    for j in range(rows - i):
      print(" ", end="")
    # Print stars
    for k in range(2 * i - 1):
      print("*", end="")
    # Print newline after each row
    print()

# Take user input for rows
try:
  rows = int(input("Enter the number of rows: "))
except ValueError:
  # Invalid input type
  print("Please enter a valid integer")
else:
  # Check if rows is positive and odd
  if rows > 0 and rows % 2 == 1:
    # Call the function with user input value
    print_hourglass(rows)
  else:
    # Invalid input for hourglass pattern
    print("Please enter a positive odd integer")
```

```
Enter the number of rows: 5
*********
 *******
  *****
   ***
    *
   ***
  *****
 *******
*********
```

13. Write a python function to print Pascal's Triangle. The number of levels in the triangle must be taken as input by the user. E.g. Input = 5 Output:

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

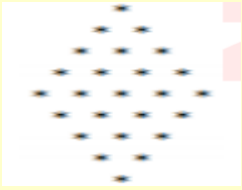
Answer:

```
def print_pascal(n):
  # initialize an empty list to store the triangle
  triangle = []
  # loop through each level of the triangle
  for i in range(n):
    # initialize an empty list to store the current row
    row = []
    # loop through each element of the current row
    for j in range(i+1):
      # if it is the first or last element, append 1
      if j == 0 or j == i:
        row.append(1)
      # otherwise, append the sum of the previous row's adjacent elements
      else:
        row.append(triangle[i-1][j-1] + triangle[i-1][j])
    # append the current row to the triangle
    triangle.append(row)
  # loop through each row of the triangle and print it with spaces
  for r in triangle:
    print(" ".join(map(str,r)))

n = int(input("Enter a number : "))
if n>0:
    print_pascal(n)
else:
    # Invalid input for hourglass pattern
    print("Please enter a positive integer")
```

```
Enter a number : 8
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
```

14. Write a python function to print Diamond Shaped Pattern shown below. Function must take integer input which represents the number of stars in the middle most line. E.g.: Input = 5 Output:



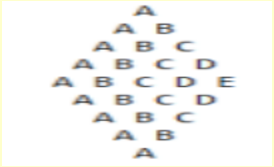
Answer:

```
def print_diamond(n):
  # check if n is odd and positive
  if n % 2 == 1 and n > 0:
    # calculate the number of rows in the upper half of the diamond
    upper_rows = (n + 1) // 2
    # loop through each row in the upper half
    for i in range(upper_rows):
      # calculate the number of spaces and stars in each row
      spaces = upper_rows - i - 1
      stars = i * 2 + 1
      # print spaces followed by stars
      print(" " * spaces + "*" * stars)
    # calculate the number of rows in the lower half of the diamond
    lower_rows = n - upper_rows
    # loop through each row in the lower half
    for j in range(lower_rows):
      # calculate the number of spaces and stars in each row
      spaces = j + 1
      stars = n - (j + 1) * 2
      # print spaces followed by stars
      print(" " * spaces + "*" * stars)

n = int(input("Enter a number : "))
if n>2:
    print_diamond(n)
else:
    # Invalid input for hourglass pattern
    print("Please enter a positive integer, greater than 2 for forming diamond")
```

```
Enter a number : 5
  *
 ***
*****
 ***
  *
```

15. Write a python function to print Diamond Shaped Character Pattern shown below. Function must take integer input within range 1 to 26, which represents the rank of the alphabet. E.g.: Input = 5 Output:



Answer:

In [21]:
```
def diamond_char(n):
  # check if n is within range 1 to 26
  if n < 1 or n > 26:
    print("Invalid input. Please enter an integer between 1 and 26.")
    return

  # create a list of alphabets from A to Z
  alphabets = [chr(i) for i in range(65,91)]

  # loop from 0 to n-1 for the upper half of the diamond
  for i in range(n):
    # print spaces before the first alphabet
    print(" " * (n-i-1), end="")
    # loop from 0 to i for the row
    for j in range(i+1):
      # print the alphabet corresponding to j
      print(alphabets[j], end="")
      # print two spaces after each alphabet except the last one
      if j < i:
        print(" " * 2, end="")
    # move to the next line
    print()

  # loop from n-2 to -1 for the lower half of the diamond
  for i in range(n-2,-1,-1):
    # print spaces before the first alphabet
    print(" " * (n-i-1), end="")
    # loop from 0 to i for the row
    for j in range(i+1):
      #printthealphabetcorrespondingtoj
      print(alphabets[j], end="  ")
    #printtwospacesaftereachalphabetexceptthelastone
    if j < i:
        print(" " *2,end="  ")
    #movetothenextline
    print()

n = int(input("Enter a number : "))

diamond_char(n)
```

```
Enter a number : 5
    A
   A  B
  A  B  C
 A  B  C  D
A  B  C  D  E
 A  B  C  D
  A  B  C
   A  B
    A
```