Python Exercises (If-Else, Loops,Functions,Arrays)

1.Print first 10 natural numbers using while loop
2.Write a python code to print the following pattern.

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

3.Write a Python program to accept a number from a user and calculate the sum of all numbers from 1 to a given number. For example, if the user entered 10, the output should be 55
4.print the multiplication table of a given number.

You can use a simple `for` loop to generate the multiplication table for a specific number.

- Set `n = 2`.
- Use `for` loop to iterate the first 10 natural numbers (From 1 to 10)
- In each iteration, multiply the current number by 2 (`p = n*i`). Now print `p`

5.Write a Python program to display only those numbers from a list that satisfy the following conditions

- •The number must be divisible by five
- •If the number is greater than 150, then skip it and move to the following number
- •If the number is greater than 500, then stop the loop

6.Write a Python program to count the total number of digits in a number using a while loop

- Set `counter = 0`
- Run while loop till `number != 0`
- In each iteration of a loop
  - Reduce the last digit from the number using floor division (`number = number // 10`)
  - Increment counter by 1
- print counter after loop execution is completed

7.Write a program to print the following start pattern using the for loop

8.The multiplication table from 1 to 10 is a table that shows the products of numbers from 1 to 10. Write a code to generates a complete **multiplication table for numbers 1 through 10**.

Use **nested loop**, where one loop is placed inside another.

- The outer loop iterates through the rows (numbers 1 to 10).
- The inner loop iterates through the columns (numbers 1 to 10) to calculate and display the product of the numbers.

```
multiplication table of: 1
 1 2 3 4 5 6 7 8 9 10
multiplication table of: 2
 2 4 6 8 10 12 14 16 18 20
multiplication table of: 3
 3 6 9 12 15 18 21 24 27 30
...
multiplication table of: 10
 10 20 30 40 50 60 70 80 90 100
```

9.print the alternate numbers pattern

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

You need to use two nested loop inside a single loop.

Think about how the numbers are arranged in each row and how that arrangement changes from one row to the next. Consider these points:

1. **Row Number:** Notice that the behavior of the numbers in a row depends on whether the row number is odd or even.
2. **Number of Elements per Row:** The first row has one number, the second has two, the third has three, and so on. The ith row contains i numbers.
3. **Odd Rows:** In odd-numbered rows (1st, 3rd, 5th, etc.), the numbers are printed in increasing order.
4. **Even Rows:** In even-numbered rows (2nd, 4th, etc.), the numbers are printed in decreasing order.
5. **Starting Number:** You need to keep track of the starting number for each row. Observe how the last number of one row relates to the first number of the next row.

Try to use these observations to build your logic row by row. You'll likely need a loop that iterates through the rows, and inside that loop, you'll have different logic based on whether the current row is odd or even.

10. Write a function `calculation()` that accepts two variables and calculates both their addition and subtraction. The function should then return both the sum and the difference in a single return statement.

11.Write a program to create a function `show_employee()` with the following specifications:

- It should accept the employee's name and salary.
- It should display both the name and salary.
- If the salary is not provided in the function call, it should default to 9000.

12.Define a function `describe_pet(animal_type, pet_name)` that prints a description of a pet. Call this function using both positional and keyword arguments.

Call the function *twice*: once by just giving the values in order, and once by saying `animal_type=` and `pet_name=`.

13.Write a recursive function to calculate the factorial of a non-negative integer.

- **Base Case:** The function needs a condition to stop calling itself. What is the factorial of 0?
- **Recursive Step:** If `n` is not 0, the factorial of `n` is `n` times the factorial of what?
- **Function Call:** The function should call *itself* with a modified argument.

14. A lambda function in Python is a small anonymous function defined using the lambda keyword. The syntax is lambda arguments: expression. The expression is evaluated and returned.

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Expected Output:

```
The even numbers in the list are: [2, 4, 6, 8, 10]
```

15.Use a lambda with the map() function to double each element in a list

Given:

```
numbers = [1, 2, 3, 4, 5]
```

Expected Output:

```
The doubled numbers are: [2, 4, 6, 8, 10]
```

16.Use a Lambda with the sorted function to sort a list of tuples based on the second element

```
data = [('apple', 5), ('banana', 2), ('cherry', 8), ('date', 1)]
```

Expected Output:

```
The sorted list of tuples based on the second element is: [('date', 1), ('banana', 2), ('
```

17.Write a Python program to guess a number between 1 and 9.
Note : User is prompted to enter a guess. If the user guesses wrong then the prompt appears again until the guess is correct, on successful guess, user will get a "Well guessed!" message, and the program will exit.

18.Write a Python program that prints all the numbers from 0 to 6 except 3 and 6.

Note : Use 'continue' statement.

Expected Output : 0 1 2 4 5

19.Write a Python program that iterates the integers from 1 to 50. For multiples of three print "Fizz" instead of the number and for multiples of five print "Buzz". For numbers that are multiples of three and five, print "FizzBuzz".

Sample Output :

fizzbuzz
1
2
fizz
4
buzz

20.Write a Python program that takes two digits m (row) and n (column) as input and generates a two-dimensional array. The element value in the i-th row and j-th column of the array should be i*j.

Note :
i = 0,1.., m-1
j = 0,1, n-1.

Test Data : Rows = 3, Columns = 4
Expected Result : [[0, 0, 0, 0], [0, 1, 2, 3], [0, 2, 4, 6]]

21.Write a Python program to check the validity of passwords input by users.

Validation :

- At least 1 letter between [a-z] and 1 letter between [A-Z].

- •At least 1 number between [0-9].
- •At least 1 character from [$#@].
- •Minimum length 6 characters.
- •Maximum length 16 characters.

22.Write a Python program to reverse the order of the items in the array.

23.Write a Python program to get the length in bytes of one array item in the internal representation.

24.Write a Python program to insert a newly created item before the second element in an existing array.

25.Write a Python program to remove a specified item using the index of an array.