

# AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh



Assignment Title:	Midterm Assignment (PiP-B)		
Assignment No:	01	Date of Submission:	15-10-2021
Course Title:	Programming in Python		
Course Code:		Section:	B
Semester:	20__-__	Course Teacher:	AKINUL ISLAM JONY

## Declaration and Statement of Authorship:

- I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
- This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgement is made.
- No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
- I/we have not previously submitted or currently submitting this work for any other course/unit.
- This work may be reproduced, communicated, compared and archived for the purpose of detecting plagiarism.
- I/we give permission for a copy of my/our marked work to be retained by the Faculty for review and comparison, including review by external examiners.
- I/we understand that Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual form, including electronic data, and oral presentations. Plagiarism occurs when the origin of them material used is not appropriately cited.
- I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or to copy my/our work.

\* Student(s) must complete all details except the faculty use part.

\*\* Please submit all assignments to your course teacher or the office of the concerned teacher.

Group Name/No.:

No	Name	ID	Program	Signature
1	RIDOY, RAYHAN KHAN	17-34755-2	BSc.CSE	
2				
3				
4				
5				
6				
7				
8				
9				
10				

## Faculty use only

FACULTY COMMENTS	Marks Obtained	
	Total Marks	

Write a program which contains a function named "list\_concat". The function receives two list and concatenate them element-wise into a new list, and then return the list. Print the final list.  
For example:  
list1 = ["py", "i", "m", "favou", "lang"]  
list2 = ["thon", "s", "y", "rite", "uage"]  
Expected output:  
["python", "is", "my", "favourite", "language"]

# answer:1

```
In [18]: import numpy as np
list1 = np.array(["py", "i", "m", "favou", "lang"])
list2 = np.array(["thon", "s", "y", "rite", "uage"])
def list_concat(l1,l2):
    new_array = np.char.add(l1, l2)
    return new_array

list_concat(list1,list2)
```

```
Out[18]: array(['python', 'is', 'my', 'favourite', 'language'], dtype='<U9')
```

From a given NumPy array create another array which contains the odd rows and even column. Finally print the new array.  
For example, for the following given array:  
arr = numpy.array([[20,66,88,12],[75,19,92,71],[27,90,33,67],[21,14,25,38],[51,44,57,77]])  
Expected output array:  
[[66 12]  
[90 67]  
[44 77]]

# answer:2

```
In [76]: arr = np.array([[20,66,88,12],[75,19,92,71],[27,90,33,67],[21,14,25,38],[51,44,57,77]])
new=arr[:,2,1::2]

print(new)

[[66 12]
 [90 67]
 [44 77]]
```

Create an 8x3 NumPy integer array from a range between 10 to 34 (using numpy.arange() function) such that the difference between each element is 1 and then Split the array into 4 (four) equal-sized sub-arrays.

# answer:3

```
In [88]: arr=np.arange(10,34,1).reshape(8,3)
print("8x3-numpy array-->\n",arr)
sub_arrays=np.split(arr,4)

print("\nfour sub arrays-->\n")
print(sub_arrays)

8x3-numpy array-->
[[10 11 12]
 [13 14 15]
 [16 17 18]
 [19 20 21]
 [22 23 24]
 [25 26 27]
 [28 29 30]
 [31 32 33]]

four sub arrays-->

[array([[10, 11, 12],
       [13, 14, 15]]), array([[16, 17, 18],
       [19, 20, 21]]), array([[22, 23, 24],
       [25, 26, 27]]), array([[28, 29, 30],
       [31, 32, 33]])]
```

Create a 3x3 NumPy integer array, and then do the following:

(a) sort the array by the second row, and

(b) sort the array by the second column.

For example,

Printing the Original array: [[34 43 73] [82 22 12] [53 94 66]]

Sorting the Original array by second row: [[73 43 34] [12 22 82] [66 94 53]]

Sorting the Original array by second column: [[82 22 12] [34 43 73] [53 94 66]]

# answer:4

```
In [94]: arr=np.random.randint(1,10,(3,3))
print(arr)
```

```
[[5 7 2]
 [5 4 5]
 [7 9 6]]
```

```
In [106]: sort_By_2ndRow = arr[:,arr[1,:].argsort()]
sort_By_2ndColumn = arr[:,arr[:,1].argsort()]

print(sort_By_2ndRow)
print("-----")
print(sort_By_2ndColumn)
```

```
[[7 5 2]
 [4 5 5]
 [9 7 6]]
```

```
-----
[[7 5 2]
 [4 5 5]
 [9 7 6]]
```

Create a parent class named Vehicle which contains three attributes ( name , mileage , capacity ), and a method called fare . fare method calculate the fare multiplying capacity by 100.  
Now, create a subclass called Bus which inherits Vehicle class. In case of a Bus , the fare charge is extra 10% with the default price because of the maintenance cost (hint: you need to override the fare method).  
Finally, create another class named Minibus which inherits Bus class. However, minibus has the fare charge half of the bus.  
Now, create object for all the classes and print the fare charges for each of the object.

### # Answer:5

```
In [36]: class Vehicle:
def __init__(self, name,mileage,capacity):
    self.name = name
    self.mileage = mileage
    self.capacity = capacity
def fare(self):
    return self.capacity*100

class Bus(Vehicle):

    def fare(self):
        bus_extra_fare=((self.capacity*100)*10)/100

        return (self.capacity*100)+bus_extra_fare

class MiniBus(Bus):

    def fare(self):
        return (super().fare())*.5

v=Vehicle("TORAG",20,34)
print(f" fare from Vehicle class: {v.fare()}")

b=Bus("ENA",30,40)
print(" fare from BUS class: ",b.fare())

m=MiniBus("DHAKAR_CHAKA",30,40)
print(f" fare from MiniBus class: {m.fare()}")
```

```
fare from Vehicle class: 3400
fare from BUS class: 4400.0
fare from MiniBus class: 2200.0
```

Create a dictionary which contains all the reserved keywords and their meanings in Python programming languag. Also, create a function to check wheter a given word (taken from user) is reserved keyword or not.

### # ANSWER:6

```
In [63]: my_dict={

    "False": "False is logical (Boolean) operations in Python",
    "True": "True is logical (Boolean) operations in Python",
    "None": "None is a special constant in Python that represents the absence of a value or a null value.",
    "and": "logical operators in Python",
    "or": "logical operators in Python",
    "not": "logical operators in Python",
    "as": "as is used to create an alias while importing a module. It means giving a different name (user-defined) to a module whi",
    "assert": "assert is used for debugging purposes.",
    "async": "provided by the asyncio library in Python. This is used to write concurrent code in Python.",
    "await": "provided by the asyncio library in Python. This is used to write concurrent code in Python.",
    "break": "break will end the smallest loop it is in and control flows to the statement immediately below the loop.",
    "continue": "continue causes to end the current iteration of the loop, but not the whole loop.",
    "class": "class is used to define a new user-defined class in Python.",
    "def": "def is used to define a user-defined function.",
    "del": "del is used to delete the reference to an object.",
    "if": "use for conditional branching or decision making.",
    "else": "use for conditional branching or decision making.",
    "elif": "use for conditional branching or decision making.",
    "except": "use with exceptions in Python",
    "raise": "use with exceptions in Python",
    "try": "use with exceptions in Python",
    "finally": "finally is used with try-except block to close up resources or file streams.",
    "for": "for is used for looping",
    "from": "from is used to import specific attributes or functions into the current namespace",
    "import": "import keyword is used to import modules into the current namespace.",
    "global": "global is used to declare that a variable inside the function is global ",
    "in": "in is used to test if a sequence (list, tuple, string etc.) contains a value. It returns True if the value is present",
    "is": "is is used in Python for testing object identity.",
    "lambda": "lambda is used to create an anonymous function",
    "nonlocal": "nonlocal is used to declare that a variable inside a nested function is not local ",
    "pass": "pass is a null statement in Python",
    "return": "return statement is used inside a function to exit it and return a value.",
    "while": "while is used for looping in Python",
    "with": "with statement is used to wrap the execution of a block of code within methods defined by the context manager.",
    "yield": "yield is used inside a function like a return statement. But yield returns a generator."

}

def check():
    user_keyword=input("ENTER A KEYWORD TO CHECK: ")

    if user_keyword in my_dict:
        print("\n YES! This is reserved keyword and meaning is ---> ",my_dict[user_keyword])
    else:
        print("\n NO! This isn't reserved keyword ")

check()
```

ENTER A KEYWORD TO CHECK: else

YES! This is reserved keyword and meaning is ---> use for conditional branching or decision making.

Write a program to create a Pascal Triangle pattern. The number of rows of the pattern will be taken from the user. Finally print the pattern.  
Hints: A pascal triangle start with "1" at the top, then continue placing numbers below it in a triangular pattern, where each number is the

numbers directly above it added together.  
A sample pattern of Pascal Triange, which contains 7 rows:  
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

**# ANSWER: 7**

```
In [67]: def printPascal(n) :  
  
    for line in range(0, n) :  
  
        for i in range(0, line + 1) :  
            print(binomialCoeff(line, i),  
                  " ", end = "")  
        print()  
  
def binomialCoeff(n, k) :  
    res = 1  
    if (k > n - k) :  
        k = n - k  
    for i in range(0, k) :  
        res = res * (n - i)  
        res = res // (i + 1)  
  
    return res  
  
n = input("Enter number to see how many rows it has:")  
n = int(n)  
printPascal(n)
```

```
Enter number to see how many rows it has:10  
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  
1 8 28 56 70 56 28 8 1  
1 9 36 84 126 126 84 36 9 1
```

In [ ]:

In [ ]: