Important Questions & Answers for GA (PDS)

1. Write a program to divide 5 by 2 so that the output is (i) 2 and (ii) 2.5

In [4]:

```
a = 5
b = 2
c = a // b
d=a/b
print('(i)',c)
print('(ii)',d)
```

```
(i) 2
(ii) 2.5
```

2. Write a python program to reverse a given string using slicing

In [3]:

```
string=input("Enter the string to revese :")
reversed_string = string[::-1]
print(reversed_string)
```

```
Enter the string to revese :PYTHON
NOHTYP
```

3. Write a program that asks the user to enter a string of at least six characters. Then print out the following:

(a) The string backwards

(b) Every character of the string except the last one

(c) Every character of the string except the first and last

(d) If the string contains a lowercase a, print out the index of the first a. Otherwise say that there is no lowercase a

(e) The string in all caps

In [2]:

```
string = input("Enter a string of at least six characters: ")
print("(a) The string backwards:", string[::-1])
print("(b) Every character of the string except the last one:", string[:-1])
print("(c) Every character of the string except the first and last:", string[1:-1])
if 'a' in string:
    print("(d) The index of the first 'a':", string.index('a'))
else:
    print("(d) There is no lowercase 'a'.")
print("(e) The string in all caps:", string.upper())
```

```
Enter a string of at least six characters: padmanabhan
(a) The string backwards: nahbanamdap
(b) Every character of the string except the last one: padmanabha
(c) Every character of the string except the first and last: admanabha
(d) The index of the first 'a': 1
(e) The string in all caps: PADMANABHAN
```

4. Write a function called reverse_only_letters() that takes a string and returns a new string where all the positions of all the alphabets have been reversed, but all other special characters are kept as it is in the position.

In [24]:

```
s=input('Enter the string to reverse :')
def reverse_only_letters(s):
    letters = [c for c in s if c.isalpha()]
    return ''.join(letters.pop() if c.isalpha() else c for c in s)
print('\nThe new string is')
reverse_only_letters(s)
```

```
Enter the string to reverse :ABCD!@#$pqrs

The new string is
```

Out[24]:

```
'srqp!@#$DCBA'
```

5. Write a program that generates and prints a list of 20 random numbers from 1 through 1000 preferebly using list comprehension. Then print out how many even numbers are in the list preferebly using filter function.

In [20]:

```
import random

random_numbers = [random.randint(1, 1000) for i in range(20)]
print("Random numbers:", random_numbers)

even_numbers = list(filter(lambda x: x % 2 == 0, random_numbers))
print("\nEven numbers:", even_numbers)

print("\nThere are", len(even_numbers), "Even numbers in the list." )
```

```
Random numbers: [848, 27, 426, 708, 831, 606, 410, 229, 434, 553, 597, 43
5, 378, 68, 996, 346, 835, 808, 914, 455]

Even numbers: [848, 426, 708, 606, 410, 434, 378, 68, 996, 346, 808, 914]

There are 12 Even numbers in the list.
```

6. Do the following:

(i) Create an array of 12 random integers using step size of 1.

(ii) Use the array created above to increase all the elements in the array by 1.

(iii) Sort the elements in the array using in-built function.

(iv) Reshape the array to 3 rows and 4 columns.

(v) Display the element in the 2nd row 3rd column.

In [14]:

```
import numpy as np

# Create an array of 12 random integers using step size of 1.
arr = np.random.randint(1, 100, 12)
print('(i)', arr)
# Use the array created above to increase all the elements in the array by 1.
arr1=arr + 1
print('(ii)',arr1)
# Sort the elements in the array using in-built function.
sorted_array=arr1.sort()
print('(iii)',sorted_array)
# Reshape the array to 3 rows and 4 columns.
reshape_array = arr.reshape(3, 4)
print('(iv)',reshape_array)
# Display the element in the 2nd row 3rd column.
print('(v)',reshape_array[1][2])
```

```
(i) [52 78 16 30 20 66 65 41 58 99 70 23]
(ii) [ 53  79  17  31  21  67  66  42  59 100  71  24]
(iii) None
(iv) [[52 78 16 30]
 [20 66 65 41]
 [58 99 70 23]]
(v) 65
```

7. Ask the user to enter a list of 10 elemnts containing numbers between 1 and 15. Then replace all of the entries in the list that are greater than 10, with 10 using map() function.

In [27]:

```
lst = eval(input("Enter a list of 10 elements having numbers between 1 & 15: "))
new_list = list(map(lambda x: x if x <= 10 else 10, lst))
print('\nThe new list is',new_list)
```

```
Enter a list of 10 elements having numbers between 1 & 15: 1,2,3,6,7,9,10,
21,13,12,15

The new list is [1, 2, 3, 6, 7, 9, 10, 10, 10, 10, 10]
```

8. Write a program to print the sum of squares of first 10 natural numbers using lambda, map and reduce.

In [7]:

```
from functools import reduce
sum_of_squares = reduce(lambda x, y: x + y, map(lambda x: x ** 2, range(1, 11)))
print(sum_of_squares)
```

```
385
```

9. Write a python program to print all letters except 'e' and 'l' in the string 'greatlearning' using while loop.

In [31]:

```python
string = "greatlearning"
print('The result of the program is\n')
i = 0
while i < len(string):
    if string[i] != 'e' and string[i] != 'l':
        print(string[i], end="")
    i += 1
```

```
The result of the program is

gratarning
```

10. Write a function called print_perfect_squares() that takes an integer and prints out all the perfect squares from 1 upto that number.

In [5]:

```python
import math
n=int(input('Enter the number to print out all the perfect squares from 1 to :'))
def print_perfect_squares(n):
    for i in range(1, n+1):
        if math.isqrt(i)**2 == i:
            print(i)
print_perfect_squares(n)
```

```
Enter the number to print out all the perfect squares from 1 to :100
1
4
9
16
25
36
49
64
81
100
```

11. Write a python program to find the sum of the Series: 1 + 1/2 + 1/3 + ... + 1/n.

In [8]:

```python
n = int(input("Enter the number of terms: "))
sum = 0
for i in range(1, n+1):
    sum = sum + (1 / i)
print("The sum of series is", round(sum, 2))
```

```
Enter the number of terms: 3
The sum of series is 1.83
```

12. Create a scientific calculator using user defined function in python. The calculator should give a menu to the user asking the type of mathematical operation to be performed (1. Addition 2. Subtraction 3. Mulitiplication and 4. Division. According to the choice of the user, operations are performed on the numbers entered. Any innovation in this basic calculator model will be given marks.

In [ ]:

```
def welcome():
    print('Welcome to Scientific Calculator!')
def calculate():
    operation = input('''
Please type in the math operation you would like to complete:
1. Addition
2. Subtraction
3. Multiplication
4. Division
''')

    x = int(input('Please enter the first number: '))
    y = int(input('Please enter the second number: '))

    if operation == '1':
        print('{} + {} = '.format(x, y))
        print(x + y)

    elif operation == '2':
        print('{} - {} = '.format(x, y))
        print(x - y)

    elif operation == '3':
        print('{} * {} = '.format(x, y))
        print(x * y)

    elif operation == '4':
        print('{} / {} = '.format(x, y))
        print(x / y)

    else:
        print('You have not typed a valid operator, please run the program again.')

def again():
        calc_again = input('''
Do you want to calculate again?
Please type Y for YES or N for NO.
''')
        if calc_again.upper() == 'Y':
            calculate()
        elif calc_again.upper() == 'N':
                print('See you later.')
        else:
            again()
welcome()
calculate()
again()
```

13. Given two lists of integers, create a third list by user defined function such that it should contain only odd numbers from the first list and even numbers from the second list? (3 marks)

- list1 = [10, 20, 23, 11, 17]
- list2 = [13, 43, 24, 36, 12]

In [3]:

```
list1 = [10, 20, 23, 11, 17]
list2 = [13, 43, 24, 36, 12]
list3=[]
for num in list1:
    if num%2!=0:
        list3.append(num)
for num in list2:
    if num%2==0:
        list3.append(num)
print('The thirt list by user defined function is',list3)
```

```
The thirt list by user defined function is [23, 11, 17, 24, 36, 12]
```

In [ ]:

```

```

14. Create a dictionary with the keys 1,2 and 3 with values 'Learning', 'For' and {A: Life, B:To , C: Great} respectively and update the value of 2 to 'Great Learning'.

In [4]:

```
my_dict = {1: 'Learning', 2: 'For', 3: {'A': 'Life', 'B': 'To', 'C': 'Great'}}
my_dict[2] = 'Great Learning'
print(my_dict)
```

```
{1: 'Learning', 2: 'Great Learning', 3: {'A': 'Life', 'B': 'To', 'C': 'Gre
at'}}
```

15. Create a 5*5 list of numbers from 1 to 10 using list comprehension. Print the nested list. And then create a dictionary whose keys are the numbers and whose values are the frequencies of the number in that nested list.

In [12]:

```
nested_list = [[j for j in range(1, 11)][i:i+5] for i in range(0, 10, 5)]

print('The nested list is \n', nested_list)

from collections import Counter

result = dict(Counter([item for sublist in nested_list for item in sublist]))

print("The dictionary whose keys are the numbers and whose values are the frequencies o
f the number in that nested list is\n",result)
```

```
The nested list is
 [[1, 2, 3, 4, 5], [6, 7, 8, 9, 10]]
The dictionary whose keys are the numbers and whose values are the frequen
cies of the number in that nested list is
 {1: 1, 2: 1, 3: 1, 4: 1, 5: 1, 6: 1, 7: 1, 8: 1, 9: 1, 10: 1}
```

16. Write a Python Program which will find all which are divisible by 3,between 4000 and 4200 ?The numbers obtained should be printed in a comma-separated sequence on a single line. (3marks)

In [23]:

```
list=[]
for i in range(4000,4201):
    if i%3==0:
        list.append(str(i))
print(','.join(list))
```

```
4002,4005,4008,4011,4014,4017,4020,4023,4026,4029,4032,4035,4038,4041,404
4,4047,4050,4053,4056,4059,4062,4065,4068,4071,4074,4077,4080,4083,4086,40
89,4092,4095,4098,4101,4104,4107,4110,4113,4116,4119,4122,4125,4128,4131,4
134,4137,4140,4143,4146,4149,4152,4155,4158,4161,4164,4167,4170,4173,4176,
4179,4182,4185,4188,4191,4194,4197,4200
```

17. Write a python program to convert a 12-hour format to a 24-hour format. The program should take the current time from the user in a 12-hour format and convert it into 24-hour format.: (5 Marks)

In [8]:

```python
def convert_to_24hour(time_str):
    # Check if last two elements are PM, then add 12 to them
    if time_str[-2:] == "pm" and time_str[:2] != "12":
        time_str = str(int(time_str[:2]) + 12) + time_str[2:]
    # Remove the AM/PM from the updated time
    if time_str[-2:] == "am" and time_str[:2] == "12":
        time_str = "00" + time_str[2:]
    # Return the formatted string
    return time_str[:-2]

time = input("Enter the current time in 12-hour format (HH:MM:SS am/pm): ")
print("The current time in 24-hour format is:", convert_to_24hour(time))
```

```
Enter the current time in 12-hour format (HH:MM:SS am/pm): 2
The current time in 24-hour format is:
```

18. Writ a Python program to Create a list of items given below

A. = [soap,detergent,chocolate,toothpaste]

B. = [soap,detergent,brush,comb,toothpaste]

C. = [juice,cake,brush,comb,chips]

Create a final dictionary that will show the total count of each product sold.

In [75]:

```python
list1 = ["soap","detergent","chocolate","toothpaste"]
list2 = ["soap","detergent","brush","comb","toothpaste"]
list3 = ["juice","cake","brush","comb","chips"]

new_list = list1 + list2 + list3

print("Original list is ", set(new_list))

countDict = dict()
for item in new_list:
    if(item in countDict):
        countDict[item] += 1
    else:
        countDict[item] = 1

print("\nPrinting count of each item \n ",countDict)
```

```
Original list is  {'cake', 'detergent', 'juice', 'comb', 'soap', 'chocolat
e', 'toothpaste', 'chips', 'brush'}

Printing count of each item
  {'soap': 2, 'detergent': 2, 'chocolate': 1, 'toothpaste': 2, 'brush': 2,
'comb': 2, 'juice': 1, 'cake': 1, 'chips': 1}
```

19. For the given list perform the following operations list_name= ['Ann', 'Pat', 'David','Tisha','Sumantha']

1. Create a list with values equal to the length of each string

2. Create a list with names of length less than 5

3. Create a dictionary with strings as key and length of the string as values

4. Sort the list with respect to the length of the string keeping the largest string at the first position

5. Find the most frequently occurring character throughout the list

In [55]:

```
list_name= ['Ann', 'Pat', 'David','Tisha','Sumantha']
list_length = [len(name) for name in list_name]
print('1. A list with values equal to the length of each string is', list_length)
new_list = [name for name in list_name if len(name) < 5]
print('2. A list with names of length less than 5 is',new_list)
dict_of_len = {element:len(element) for element in list_name}
print('3. A dictionary with strings as key and length of the string as values is \n',di
ct_of_len)
list_name.sort(key=len, reverse=True)
print('4. Sorted list is',list_name)
```

```
1. A list with values equal to the length of each string is [3, 3, 5, 5,
8]
2. A list with names of length less than 5 is ['Ann', 'Pat']
3. A dictionary with strings as key and length of the string as values is
 {'Ann': 3, 'Pat': 3, 'David': 5, 'Tisha': 5, 'Sumantha': 8}
4. Sorted list is ['Sumantha', 'David', 'Tisha', 'Ann', 'Pat']
```

In [42]:

```
list_name = ['Ann', 'Pat', 'David','Tisha','Sumantha']
most_frequent_char = max(list_name, key = list_name.count)
print("5. The most frequently occurring character throughout the list is:", most_freque
nt_char)
```

```
5. The most frequently occurring character throughout the list is: Ann
```

20. Write a user-defined function to solve the given task. From the given vehicle number plates determine are these numbers even or odd. Output the dictionary with key as number and value as (even/odd) as appropriate

number_plates = ['MH 12 XJ - 2234', 'UP 04 LG - 2455', 'GJ 34 RV - 2442', 'KL 07 AP-2433']

In [56]:

```
number_plates = ['MH12XJ-2234', 'UP04LG-2455', 'GJ34RV-2442', 'KL07AP-2433']
def even_odd_number(number_plates):
    result = {}
    for number in number_plates:
        number = number[-1]
        if int(number) % 2 == 0:
            result[number] = "even"
        else:
            result[number] = "odd"
    return result

print(even_odd_number(number_plates))
```

```
{'4': 'even', '5': 'odd', '2': 'even', '3': 'odd'}
```

21. Writ a Python function that will print out the multiplication table for any number you give it.

In [58]:

```
n=int(input('Enter the number to display its multiplication table :'))
def multiplication_table(n):
    for i in range(1, 11):
        print(n, 'x', i, '=', n*i)
multiplication_table(n)
```

```
Enter the number to display its multiplication table :5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

22. Write a program to print the following pattern.

*

*

*

*

*

In [70]:

```
for i in range(5):
    for j in range(5):
        print('*',end=" ")
    print()
```

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

23. Write a program to print the following pattern.

*

*

*

In [71]:

```
for i in range(5):
    for j in range(i+1):
        print('*',end=" ")
    print()
```

```
*
* *
* * *
* * * *
* * * * *
```

24. Write a program to print the following pattern.

*

*

*

In [72]:

```
for i in range(5):
    for j in range(5-i):
        print('*',end=" ")
    print()
```

```
* * * * *
* * * *
* * *
* *
*
```

25. Write a program to print the following number pattern.

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

In [69]:

```
for i in range(6):
    for j in range(1,i+1):
        print(j,end=" ")
    print()
```

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

26. Write a program to print the following number pattern.

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

In [66]:

```
for i in range(6):
    for j in range(i):
        print(i,end=" ")
    print("\n")
```

```
1 

2 2 

3 3 3 

4 4 4 4 

5 5 5 5 5 
```

In [ ]:

```

```