**Simple practice problems**

**To be added to the Final Exam**

|  |  |
| --- | --- |
| **1** | **Write a Python program to converting an integer to a string in any base.**  **Sample Solution**:  **Python Code:**  **def to\_string(n,base):**  **conver\_tString = "0123456789ABCDEF"**  **if n < base:**  **return conver\_tString[n]**  **else:**  **return to\_string(n//base,base) + conver\_tString[n % base]**  **print(to\_string(2835,16))** |
| **2** | **Write a Python program to solve the Fibonacci sequence using recursion.**  **Sample Solution**:  **Python Code:**  **def fibonacci(n):**  **if n == 1 or n == 2:**  **return 1**  **else:**  **return (fibonacci(n - 1) + (fibonacci(n - 2)))**  **print(fibonacci(7))** |
| **3** | **Write a Python program to calculate the sum of the positive integers of n+(n-2)+(n-4)... (until n-x =< 0). // use recursion**  **Sample Solution:**  **Python Code:**  **def sum\_series(n):**  **if n < 1:**  **return 0**  **else:**  **return n + sum\_series(n - 2)**  **print(sum\_series(6))**  **print(sum\_series(10))** |
| **4** | **Write a Python program to calculate the value of 'a' to the power 'b'.**  **Test Data: (power(3,4) -> 81**  **Sample Solution:-**  **Python Code:**  **def power(a,b):**  **if b==0:**  **return 1**  **elif a==0:**  **return 0**  **elif b==1:**  **return a**  **else:**  **return a\*power(a,b-1)**  **print(power(3,4))** |
| **5** | **Write a Python program to find out its proper divisors**  e.g.  input 12  output : 1, 2, 3, 4 , 6  **# to print all divisors**  **Python Code:**  **# method to print the divisors**  **def printDivisors(n) :**  **i = 1**  **while i <= n :**  **if (n % i==0) :**  **print i,**  **i = i + 1** |
| **6** | **Write a Python program to multiply two integers without using the \* operator in python.**  **Sample Solution:-**  **Python Code:**  **def multiply(x, y):**  **if y < 0:**  **return -multiply(x, -y)**  **elif y == 0:**  **return 0**  **elif y == 1:**  **return x**  **else:**  **return x + multiply(x, y - 1)**  **print(multiply(3, 5));** |
| **7** | **Python program to display the given integer in reverse manner**  **Python Code:**  **number = int(input("Enter a positive integer: "))**  **rev = 0**  **while(number!=0):**  **digit = number%10**  **rev = (rev\*10)+digit**  **number = number//10**  **print(rev)** |
| **8** | **Python program to find the sum of the digits of an integer using while loop**  **Python Code:**  **sum = 0**  **number = int(input("Enter an integer: "))**  **while(number!=0):**  **digit = number%10**  **sum = sum+digit**  **number = number//10**  **print("Sum of digits is: ", sum)** |
| **9** | **Write a program in python as a function that determines the intersection of two sets without using any predefined functions.**  **Python Code:**  **def set\_intersect(E, F):**  **# initialising the intersection of E and F**  **E\_inter\_F = set({})**  **for x in E:**  **if x in F:**  **E\_inter\_F.add(x)**  **return E\_inter\_F**  **# Testing algorithm**  **E = {"Java", "Python", "Javascript", "C ++", "C #"}**  **F = {"VB.ET", "Java", "Kotlin", "Python"}**  **print("The intersection of E and F is : " , set\_intersect(E , F))**  **# The output is : The intersection of E and F is : {'Java', 'Python'}** |
| **10** | **Write a python algorithm to remove duplicate elements from a list.**  **Solution**  **Python Code:**  **def removeDuplicate(L):**  **# initializing the list without duplicate elements**  **unique = []**  **for x in L:**  **if x not in unique:**  **unique.append(x)**  **return unique**  **# testing the algorithm**  **L = [11 , 3 , 11, 3 , 4 , 11 , 7 , 3 , 11]**  **print(removeDuplicate(L)) # the output is : [11, 3, 4, 7]** |
| **11** | **write a Python program that calculates two sets' union without using any inbuilt function like union()?**  **Method 1:**  **set1 = {2, 4, 5, 6}**  **set2 = {4,6, 7, 8}**  **lst = {i: i for i in list(set1) + list(set2)}.values()**  **lst={item for item in lst}**    **print(lst)**  **Method 2:**  **set1 = {2, 4, 5, 6}**  **set2 = {4, 6, 7, 8}**  **lst = list(set1)**  **for item in set2:**  **if item not in lst:**  **lst.append(item)**  **lst={item for item in lst}**    **print(lst)** |