Name of the Paper	**:**	**Soft Computing**
Lab Paper Code	**:**	**BCA692B**
Registration No.	**:**	**18013000073 of 2018-2019**
Student Name	**:**	**ABHIJIT BARIK**
University Roll Number	**:**	**18010301001**
Course	**:**	**BCA**
Semester	**:**	**6th**
Department	**:**	**Computational Science**

ASSIGNMENT -1

Assignment on basic Python

1.write a program in python to find all the prime numbers from 1 to 100

In [3]:
```
lower = 1
upper = 100

print("Prime numbers between", lower, "and", upper, "are:")

for num in range(lower, upper + 1):
   # all prime numbers are greater than 1
   if num > 1:
       for i in range(2, num):
           if (num % i) == 0:
               break
       else:
           print(num)
```

```
Prime numbers between 1 and 100 are:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

2. Write a program in python to check whether a number is ARMSTRONG or not.

In [7]:
```
# take input from the user
num = int(input("Enter a number: "))
order = len(str(num))

# initialize sum
sum = 0

# find the sum of the cube of each digit
temp = num
while temp > 0:
   digit = temp % 10
   sum += digit ** order
   temp //= 10

# display the result
if num == sum:
   print(num,"is an Armstrong number")
else:
   print(num,"is not an Armstrong number")
```

```
Enter a number: 153
153 is an Armstrong number
```

3. Write a program in python to calculate greatest common divisor of a number

In [9]:
```
def hcf(x, y):

    # Get the minimum of two numbers
    if x > y:
        smaller = y
    else:
        smaller = x

    # Using a for loop to checking the remainder
    for i in range(1, smaller + 1):
        if ((x % i == 0) and (y % i == 0)):
            # If all are zero, then assign
            hcf = i
    return hcf

# Enter two numbers
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))

print("The greatest common divisor: ", hcf(num1, num2))
```

```
Enter the first number: 12
Enter the second number: 14
The greatest common divisor:  2
```

4. Create a list in python for food items: perform the following tasks on the list a) Append a new item "Daal Makhni" at the end of the list b) Add a new item in the 4th position c) Delete last two items from the list d) Sort the list according to alphabetical order e) Traverse the list and print f) Print the list selectively(Every alternative item) g) Print the list in reverse order

In [22]:
```
print("create a list\n")


listdata = ["Mango", "apple", "banana", "sugarcane"]
print(listdata,"\n")
print("Append a new item "Daal Makhni" at the end of the list\n ")
listdata.append("Daal Makhni")
print(listdata,"\n")

print("Add a new item in the 4th position\n")
listdata.insert(3,'Orange')
print(listdata,"\n")
print("Sort the list according to alphabetical order\n")
listdata.sort()
print(listdata,"\n")
print("Traverse the list and print\n")
for i in range(0, len(listdata)):
   print(listdata[i],"\n")
print("Print the list selectively(Every alternative item)\n")

for count in listdata[::2]:
    print(count)


print("Print the list in reverse order\n")
print(listdata[::-1])
```

```
create a list

['Mango', 'apple', 'banana', 'sugarcane']

Append a new item "Daal Makhni" at the end of the list

['Mango', 'apple', 'banana', 'sugarcane', 'Daal Makhni']

Add a new item in the 4th position

['Mango', 'apple', 'banana', 'Orange', 'sugarcane', 'Daal Makhni']

Sort the list according to alphabetical order

['Daal Makhni', 'Mango', 'Orange', 'apple', 'banana', 'sugarcane']

Traverse the list and print

Daal Makhni

Mango

Orange

apple

banana

sugarcane

Print the list selectively(Every alternative item)

Daal Makhni
Orange
banana
Print the list in reverse order

['sugarcane', 'banana', 'apple', 'Orange', 'Mango', 'Daal Makhni']
```

5. Create a nested list of students having different lists for different class tests (class test 1 and class test 2). Print the marks of the students in ascending order.

In [12]:
```
classtest1=[20,30,50]
classtest2=[70,60,100]
classtest1.sort()
classtest2.sort()
classtest=[classtest1,classtest2]
for item in classtest:
        for x in item:
            print(x)
        print("\n")
```

```
20
30
50


60
70
100
```

6. Create a dictionary in python for displaying information of employee (Emp ID, Emp name, Dept, Designation, Salary). Do the following tasks with the dictionary: a) Add at least 6 member in emp list b) Add a new employee details c) Add a new field called incentives d) Select the list of employee who is/are getting incentives more than 1000 rs e) Remove an employee from HR department f) Select the name of employee who is getting maximum salary g) Select the name and salary of employee , department wise

In [15]:
```
emp={'Emp ID':[], 'Emp name':[],'Dept':[],'Designation':[],'salary':[]}
print(emp)
print(" Add at least 6 member in emp list \n")
emp['Emp name'].append('Avijit')
emp['Emp name'].append('Asok')
emp['Emp name'].append('Priya')
emp['Emp name'].append('Hardik')
emp['Emp name'].append('Alok')
emp['Emp name'].append('Anik')
print(emp)
print(" After adding new employee details\n ")
emp['Emp ID'].append(1)
emp['Dept'].append('IT')
emp['Designation'].append('Associate')
emp['salary'].append(60000)
print(emp)
#Add a new field
print("After adding new field \n")
emp['incentives']=1000

print(emp)
```

```
{'Emp ID': [], 'Emp name': [], 'Dept': [], 'Designation': [], 'salary': []}
 Add at least 6 member in emp list

{'Emp ID': [], 'Emp name': ['Avijit', 'Asok', 'Priya', 'Hardik', 'Alok', 'Anik'], 'Dept': [], 'Designation': [], 'salary': []}
 After adding new employee details

{'Emp ID': [1], 'Emp name': ['Avijit', 'Asok', 'Priya', 'Hardik', 'Alok', 'Anik'], 'Dept': ['IT'], 'Designation': ['Associate'], 'salary': [60000]}
After adding new field

{'Emp ID': [1], 'Emp name': ['Avijit', 'Asok', 'Priya', 'Hardik', 'Alok', 'Anik'], 'Dept': ['IT'], 'Designation': ['Associate'], 'salary': [60000], 'incentives': 1000}
```

7. Create a nested dictionary on python(Student – BCA/MCA) Perform the following tasks on the nested dictionary: a) Access the element from nested dictionary b) Merge two dictionary c) Remove two items from nested dictionary d) Remove last inserted item e) Add an element in a specific position f) Remove last inserted item from the dictionary

In [43]:
```
Student = { 'BCA': {1: 'Avi',2:'Alok',3:'Risav'},
                   'MCA': {1: 'Anu',2:'priya',3:'Asok'}
                  }

print("After creating nested dictionary\n",Student)

print("Access the element from nested dictionary\n")
print(Student['BCA'][1],Student['MCA'][2])

# Python code to merge dict using update() method
def Merge(dict1, dict2):
        return(dict2.update(dict1))


dict1 = {'a': 10, 'b': 8}
dict2 = {'d': 6, 'c': 4}

# This return None
Merge(dict1, dict2)

# changes made in dict2
print("After merging")
print(dict2)
print("After deletion")
del dict2['d']
print(dict2)
dict2.popitem()
print("Remove last inserted item\n")
print(dict2)
```

```
After creating nested dictionary
 {'BCA': {1: 'Avi', 2: 'Alok', 3: 'Risav'}, 'MCA': {1: 'Anu', 2: 'priya', 3: 'Asok'}}
Access the element from nested dictionary

Avi priya
After merging
{'d': 6, 'c': 4, 'a': 10, 'b': 8}
After deletion
{'c': 4, 'a': 10, 'b': 8}
Remove last inserted item

{'c': 4, 'a': 10}
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: