1) #python program to illustrate various operations on primitive data type

int1=20

int2=50

#operations on integers

print(int1+int2)

print(int1-int2)

print(int1\*int2)

print(int1/int2)

#operation on strings

string\_1=("im vaibhav")

string\_2=(" from hubli")

print(string\_1+string\_2)

#operations on float

float1=11.2

float2=55.3

print(float1+float2)

print(float1-float2)

print(float1\*float2)

print(float1/float2)

#operation on boolean

marks=80

if(marks>=50):

print("passed")

else:

print("fail")

**OUTPUT:-**

70

-30

1000

0.4

im vaibhav from hubli

66.5

-44.099999999999994

619.3599999999999

0.20253164556962025

Passed

**OUTPUT 2:-**

130

-70

3000

0.3

im vaibhav from hubli

77.19999999999999

-39.4

1101.87

0.3241852487135506

Fail

**OUTPUT:-3**

180

-20

8000

0.8

im vaibhav from hubli

79.1

-38.5

1193.6399999999999

0.3452380952380953

Passed

**OUTPUT 4:-**

20

0

100

1.0

im vaibhav from hubli

100.6

-60.0

1630.09

0.25280199252801994

Fail

**OUTPUT 5:-**

60

-40

500

0.2

im vaibhav STUDYING IN DIPLOMA

65.6

-24.999999999999996

919.5899999999999

0.44812362030905084

Passed

2#python program to implement primitive data type

import array as arr

A=arr.array('i', [])

N=int(input("enter the array size="))

for i in range(N):

x=int(input("enter the array of (i) element="))

A.append(x)

print("array elements=")

for i in range(N):

print(A[i])

**OUTPUT:-**

enter the array size=5

enter the array of (i) element=1

enter the array of (i) element=2

enter the array of (i) element=9

enter the array of (i) element=6

enter the array of (i) element=8

array elements=

1

2

9

6

8

**OUTPUT 2:-**

enter the array size=2

enter the array of (i) element=7

enter the array of (i) element=5

array elements=

7

5

**OUTPUT:-3**

enter the array size=3

enter the array of (i) element=8

enter the array of (i) element=7

enter the array of (i) element=6

array elements=

8

7

6

**OUTPUT:-4**

enter the array size=8

enter the array of (i) element=7

enter the array of (i) element=5

enter the array of (i) element=6

enter the array of (i) element=9

enter the array of (i) element=8

enter the array of (i) element=4

enter the array of (i) element=2

enter the array of (i) element=1

array elements=

7

5

6

9

8

4

2

1

**OUTPUT:-5**

enter the array of (i) element=5

array elements=5

3#write a python program to implement abstract data type

class Student:

def getstudentdetails(self):

self.rollno=input("enter roll number=")

self.name=input("enter name=")

self.os=int(input("enter os marks="))

self.ds=int(input("enter ds marks="))

self.se=int(input("enter se marks="))

def printresult(self):

self.persentage=(int)((self.ds+self.os+self.se)/300\*100);

print(self.rollno,self.name,self.persentage,"%")

s1=Student()

s1.getstudentdetails()

print("results=")

s1.printresult()

s1.ds+=9

print("after adding ia marks=")

s1.printresult()

**OUTPUT 1:-**

enter roll number=59

enter name=vaibhav

enter os marks=96

enter ds marks=45

enter se marks=82

results=

59 vaibhav 74 %

after adding ia marks=

59 vaibhav 77 %

**OUTPUT 2:-**

enter roll number=96

enter name=SHASHI

enter os marks=85

enter ds marks=75

enter se marks=69

results=

96 SHASHI 76 %

after adding ia marks=

96 SHASHI 79 %

**OUTPUT 3:-**

enter roll number=75

enter name=NAGU

enter os marks=95

enter ds marks=91

enter se marks=93

results=

75 NAGU 93 %

after adding ia marks=

75 NAGU 96 %

**OUTPUT 4:-**

enter roll number=80

enter name=RITISH

enter os marks=85

enter ds marks=75

enter se marks=45

results=

80 RITISH 68 %

after adding ia marks=

80 RITISH 71 %

**OUTPUT 5:-**

enter roll number=1

enter name=GOUTAM

enter os marks=10

enter ds marks=20

enter se marks=30

results=

1 GOUTAM 20 %

after adding ia marks=

1 GOUTAM 23 %

4#python program to implement linear search

#define list

A=[]

#take input list n

n=int(input("enter the size of array="))

for i in range(n):

x=int(input("enter {i} element="))

A.append(x)

S=int(input("enter number to search in list="))

flag=0

for i in range(n):

if S==A[i]:

flag=1

break

if flag==1:

print(f"{S}was found at index {i}")

else:

print(f"{S} was not found")

**OUTPUT 1:-**

enter the size of array=5

enter {i} element=4

enter {i} element=8

enter {i} element=5

enter {i} element=8

enter {i} element=9

enter number to search in list=10

10 was not found

**OUTPUT 2:-**

enter the size of array=4

enter {i} element=8

enter {i} element=6

enter {i} element=5

enter {i} element=8

enter number to search in list=8

8was found at index 0

**OUTPUT 3:-**

enter the size of array=5

enter {i} element=4

enter {i} element=8

enter {i} element=6

enter {i} element=9

enter {i} element=3

enter number to search in list=5

5 was not found

**OUTPUT 4:-**

enter the size of array=2

enter {i} element=4

enter {i} element=8

enter number to search in list=5

5 was not found

**OUTPUT 5:-**

enter the size of array=5

enter {i} element=4

enter {i} element=5

enter {i} element=6

enter {i} element=5

enter {i} element=3

enter number to search in list=5

5 was found at index 1

5#python program to implement binary search

def binarysearch(list,low,high,s):

if high>=low:

mid=(high+low)//2

if list[mid]==s:

return mid

elif list[mid]>s:

return binarysearch(list,low,mid-1,s)

else:

return binarysearch(list,mid+1,high,s)

else:

return -1

#define list

lst=[]

size=int(input("enter the size of list="))

for i in range(size):

n=int(input(f"enter list of {i} elements="))

lst.append(n)

s=int(input("enter element to search="))

result=binarysearch(lst,0,len(lst)-1,s)

if result!=-1:

print("element is found at index",str(result))

else:

print("element is not found")

**OUTPUT 1:-**

enter the size of list=5

enter list of 0 elements=1

enter list of 1 elements=2

enter list of 2 elements=3

enter list of 3 elements=4

enter list of 4 elements=5

enter element to search=2

element is found at index 1

**OUTPUT 2:-**

enter the size of list=2

enter list of 0 elements=4

enter list of 1 elements=5

enter element to search=2

element is not found

**OUTPUT 3:-**

enter the size of list=6

enter list of 0 elements=1

enter list of 1 elements=2

enter list of 2 elements=3

enter list of 3 elements=2

enter list of 4 elements=4

enter list of 5 elements=5

enter element to search=2

element is found at index 3

**OUTPUT 4:-**

enter the size of list=3

enter list of 0 elements=1

enter list of 1 elements=2

enter list of 2 elements=3

enter element to search=20

element is not found

**OUTPUT 5:-**

enter the size of list=4

enter list of 0 elements=10

enter list of 1 elements=50

enter list of 2 elements=330

enter list of 3 elements=42

enter element to search=27

element is not found

6#python program to implement bubble sort

a=[]

n=int(input("enter the size of elements"))

for i in range(n):

value=int(input("enter {i} elements="))

a.append(value)

for i in range(n-1):

for j in range(n-i-1):

if(a[j]>a[j+1]):

temp=a[j]

a[j]=a[j+1]

a[j+1]=temp

print("the result in ascending order=",a)

**OUTPUT 1:-**

enter the size of elements5

enter {i} elements=8

enter {i} elements=2

enter {i} elements=6

enter {i} elements=9

enter {i} elements=4

the result in ascending order= [2, 4, 6, 8, 9]

**OUTPUT 2:-**

enter the size of elements5

enter {i} elements=20

enter {i} elements=2

enter {i} elements=10

enter {i} elements=9

enter {i} elements=8

the result in ascending order= [2, 8, 9, 10, 20]

**OUTPUT 3:-**

enter the size of elements5

enter {i} elements=8

enter {i} elements=2

enter {i} elements=5

enter {i} elements=9

enter {i} elements=4

the result in ascending order= [2, 4, 5, 8, 9]

**OUTPUT 4:-**

enter the size of elements4

enter {i} elements=8

enter {i} elements=2

enter {i} elements=6

enter {i} elements=9

the result in ascending order= [2,6, 8, 9]

**OUTPUT 5:-**

enter the size of elements6

enter {i} elements=8

enter {i} elements=2

enter {i} elements=6

enter {i} elements=9

enter {i} elements=4

enter {i} elements=10

the result in ascending order= [2, 4, 6, 8, 9,10]