INDIAN INSTITUTE OF TECHNOLOGY GANDHINAGAR

Problem Sheet

ES-102 and ES-112

1. Write a program that examines three variables x, y, and z and prints the largest odd number among them. If none of them are odd, it should print a message to that effect.

2. Write a program that asks the user to input 10 integers, and then prints the largest odd number that was entered. If no odd number was entered, it should print a message to that effect.

3. Write a program that asks the user to enter an integer and prints two integers, root and pwr, such that $0 < \text{pwr} < 6$ and root**pwr is equal to the integer entered by the user. If no such pair of integers exists, it should print a message to that effect.

4. Let s be a string that contains a sequence of decimal numbers separated by commas, e.g., s = '1.23,2.4,3.123'. Write a program that prints the sum of the numbers in s.

5.
```
x = 25
epsilon = 0.01
numGuesses = 0
low = 0.0
high = max(1.0, x)
ans = (high + low)/2.0
while abs(ans**2 - x) >= epsilon:
    print 'low =', low, 'high =', high, 'ans =', ans
    numGuesses += 1
    if ans**2 < x:
      low = ans
    else:
      high = ans
    ans = (high + low)/2.0
print 'numGuesses =', numGuesses
print ans, 'is close to square root of', x
```

(i) What would the above code do if the statement x = 25 were replaced by x = -25?

(ii) What would have to be changed to make the above code work for finding an approximation to the cube root of both negative and positive numbers?

(Hint: think about changing low to ensure that the answer lies within the region being searched.)

6. What is the decimal equivalent of the binary number 10011?

7.
```
x = 0.0
for i in range(10):
    x = x + 0.1
    if x == 1.0:
        print x, '= 1.0'
    else:
        print x, 'is not 1.0'
print 1.0 is not 1.0
```

Above code gives a very unexpected answer, give reasons.

8. Write a function isIn that accepts two strings as arguments and returns True if either string occurs anywhere in the other, and False otherwise.

(Hint: you might want to use the built-in str operation **in**.)

9. What on earth is happening here ?

```
def f(x):
    def g():
        x = 'abc'
        print 'x =', x
    def h():
        z = x
        print 'z =', z
    x = x + 1
    print 'x =', x
    h()
    g()
    print 'x =', x
    return g
x = 3
z = f(x)
print 'x =', x
print 'z =', z
z()
```

Run and Analyze.