1. What is the difference between 'is' operator and '==' operator? Explain with an example.

'is' and equals(==) operators are mostly same but they are not same. 'is' operator defines if both the variables point to the same object whereas the == sign checks if the values for the two variables are the same.

```
list1 = []
list2 = []
list1=list2

if (list1 == list2):
    print("True")
else:
    print("False")
if (list1 is list2):
    print("True")
else:
    print("False")
if (list1 is list3):
    print("True")
else:
    print("False")
```

```
True
False
True
```

2. Convert the below binary numbers into decimal.

a) 10100011

b) 101101

c) 1101010101010

```
# using built-in int() function
print(int('10100011',2))
print(int('101101',2))
print(int('1101010101010',2))
```

```
163
45
3370
```

Manually converting to decimal

```
bin_num = '10100011'
position = len(bin_num) - 1
int_num = 0
for i in bin_num:
    int_num += int(i) * (2 ** position)
    position -= 1
print(int_num)
```

```
163
```

Decimal Position	7	6	5	4	3	2	1	0
Binary number	1	0	1	0	0	0	1	1
Decimal equivalent number to each position	128	0	32	0	0	0	2	1
Total	163							

```
bin_num = '101101'
position = len(bin_num) - 1
int_num = 0
for i in bin_num:
    int_num += int(i) * (2 ** position)
    position -= 1
print(int_num)
```

```
45
```

Decimal Position	5	4	3	2	1	0
Binary number	1	0	1	1	0	1
Decimal equivalent number to each position	32	0	8	4	0	1
Total	45					

```
bin_num = '1101010101010'
position = len(bin_num) - 1
int_num = 0
for i in bin_num:
    int_num += int(i) * (2 ** position)
    position -= 1
print(int_num)
```

```
3370
```

Decimal Position	11	10	9	8	7	6	5	4	3	2	1	0
Binary number	1	1	0	1	0	1	0	1	0	1	0	0
Decimal equivalent number to each position	2048	1024	0	256	0	0	32	0	8	0	2	0
Total	3370											

3. Convert the below decimal numbers into binary.

a) 239

b) 66

```
# using built-in bin() function
print(bin(239))
print(bin(66))
```

```
0b11101111
0b1000010
```

Manually converting to binary

Divisor	Dividend	Modulo
2	239	
2	119	1
2	59	1
2	29	1
2	14	1
2	7	0
2	3	1
	1	1

So the binary version is 11101111

```
# converting using python
int_num = 239
quotient = int_num
module = []
while quotient != 1:
    module.append(quotient%2)
    quotient = quotient // 2
module.append(1)
module.reverse()
binary_num = int("".join(str(num) for num in module))
binary_num
```

```
11101111
```

Divisor	Dividend	Modulo
2	66	
2	33	0
2	16	1
2	8	0
2	4	0
2	2	0
	1	0

So the binary version is 1000010

```
int_num = 66
quotient= int_num
module = []
while quotient != 1:
    module.append(quotient%2)
    quotient = quotient // 2
module.append(1)
module.reverse()
binary_num = int("".join(str(num) for num in module))
binary_num
```

```
1000010
```

4. Write a program that prints the integers from 1 to 100. But for multiples of three print "Fizz" instead of the number, and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

```
for i in range(1,1054):
    if (i%3 == 0) & (i%5 == 0):
        print("FizzBuzz")
    elif (i%3 == 0):
        print("Fizz")
    elif (i%5 == 0):
        print("Buzz")
    else:
        print(i)
```

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
Fizz
22
23
Fizz
Buzz
26
Fizz
28
29
FizzBuzz
31
32
Fizz
34
Buzz
Fizz
37
38
Fizz
Buzz
41
Fizz
43
44
FizzBuzz
46
47
Fizz
49
Buzz
Fizz
52
53
Fizz
Buzz
56
Fizz
58
59
FizzBuzz
61
62
Fizz
64
Buzz
Fizz
67
68
Fizz
Buzz
71
Fizz
73
74
FizzBuzz
76
77
Fizz
79
Buzz
Fizz
82
83
Fizz
Buzz
86
Fizz
88
89
FizzBuzz
91
92
Fizz
94
Buzz
Fizz
97
98
Fizz
Buzz
```

5. Declare two integers. Print the lesser of two given numbers if both numbers are even, but print the greater number if one or both the numbers are odd.

```
def compare(num1, num2):
    if (num1%2 == 0) & (num2%2 == 0):
        print(min(num1,num2))
    else:
        print(max(num1,num2))
```

```
compare(40,50)
```

```
40
```

```
compare(603,5003)
```

```
5003
```

```
compare(60,5003)
```

```
5003
```

6. Declare a two-word string separated by space. Print True if both the words begin with the same letter (case-sensitive) otherwise print False.

```
# input 1
input_string = 'Hello World'
words_list = input_string.split(' ')
if words_list[0][0] == words_list[1][0]:
    print('True')
else:
    print('False')
```

```
False
```

```
# input 2
input_string = 'Jumping Jack'
words_list = input_string.split(' ')
if words_list[0][0] == words_list[1][0]:
    print('True')
else:
    print('False')
```

```
True
```

```
# input 3
input_string = 'Jumping Jack'
words_list = input_string.split(' ')
if words_list[0][0].lower() == words_list[1][0].lower():
    print('True')
else:
    print('False')
```

```
True
```

7. Enhance question #6 to include case-insensitive comparison.

```
input_string = 'Jumping Jack'
words_list = input_string.split(' ')
if words_list[0][0].lower() == words_list[1][0].lower():
    print('True')
else:
    print('False')
```

```
True
```

8. Use 'for', '.split()', and 'if' to print out words that start with 'b'(case-sensitive).

```
input_string = 'You cannot end a sentence with because because because is a conjunction.'
word_list = input_string.split(' ')
for word in word_list:
    if word[0] == 'b':
        print(word)
```

```
because
because
because
```

9. Enhance question #8 to include case-insensitive comparison and remove duplicates from the output.

```
input_string = 'You cannot end a sentence with because because because is a conjunction.'
word_list = list(dict.fromkeys(input_string.split(' ')))
# Alternate solution
# word_list = list(set(input_string.split(' ')))
for word in word_list:
    if word[0].lower() == 'b':
        print(word)
```

```
because
because
```

10. Write a Python program to swap two variables.

```
first_num = 10
second_num = 20
first_num, second_num = second_num, first_num
print('first_num =',first_num)
print('second_num =',second_num)
```

```
first_num = 20
second_num = 10
```

Bonus

1. Given a list of integers, return indices of the two numbers such that they add up to a specific target.

```
nums = [2,7,11,15]
target = 9
index_combination=[]
for i in range(len(nums)):
    difference = target - nums[i]
    try:
        if nums.index(difference, i+1):
            index_combination += [i, nums.index(difference, i+1)]
            print(index_combination)
            break
    except:
        continue
if len(index_combination) ==0:
    print("No combination found")
```

```
[0, 1]
```

2. Without using swapcase(), You are given a string and your task is to swap cases. In other words, convert all lowercase letters to uppercase letters and vice versa.

```
input_string = "McDonald's"
output_string = ""
for letter in input_string:
    if letter.islower():
        output_string += letter.upper()
    else:
        output_string += letter.lower()
print(output_string)
```

```
mCdONALD'S
```

3. Write a Python program to convert decimal number to binary.

```
decimal_num = 239
quotient= decimal_num
module = []
while quotient != 1:
    module.append(quotient%2)
    quotient = quotient // 2
module.append(1)
module.reverse()
binary_num = int("".join(str(num) for num in module))
binary_num
```

```
11101111
```