```
#1)Function to implement "if" statement
def if_function(condition, true_result, false_result):
    if condition == True:
        return true_result
    else:
        return false_result

def with_if_statement():
    if c():
        return t()
    else:
        return f()

def with_if_function():
    return if_function(c(), t(), f())

def c():
    return False

def t():
    1/0

def f():
    return 1


print(if_function(False, 2, 3))
```

```
#2)Function to add all the odd number given in a range

num = int(input("Enter a number:"))

def sum_odd(num):
    odd = 0
    for i in range(1, num+1):
        if(i % 2 == 1):
            odd = odd + i
    print(odd)

sum_odd(num)
```

```
# 3) Add square of two smallest number
print("Enter 4 numbers: ")
a = input()
b=input()
c=input()
d=input()
e = [a,b,c,d]
e.sort()
print("Num: ", e[0],e[1])
a= int(e[0])
b=int(e[1])
sum_odd = (a**2 + b**2)

print("The sum of the two smallest numbers is: ", sum_odd)

# 4)Write a function named "df" that takes three integers x, y, and z. It
returns whether
# subtracting one of these numbers from another gives the third


def is_true(x, y, z):
   while (x and y and z):
       a = y - z
       b = x - z
       c = x - y
       if (a == x or b == y or c == z):
           return print("True")
       break
   else:
       return print("False")

x = 5
y = 3
z = 2

print(is_true(x, y, z))


# 5) Create a function that takes an integer m greater than 1 and returns the
largest integer
# smaller than m that evenly divides m.
x =[]
def lrgst_factor(m):
   print("the factors of", m, "are: ")
   for i in range(1, m + 1):
       if m % i == 0 and i < m:
           x.append(int(i))
   print(x)
   print(max(x))
```

```
num = 80
print(lrgst_factor(num))


# 6) Define a function which takes in a number n and determines whether the
number is a
# perfect number.

x =[]
def pfct_num(n):
   print("the factors of", n, "are: ")
   for i in range(1, n + 1):
       if n % i == 0 and i < n:
           x.append(int(i))
   print(x)
   print(max(x))
   if sum(x) == n:
       print("True")
   else:
       print("False")

num = 6
print(pfct_num(num))


# 7) Implement a function to check if the number of bits from two positive
input parameters is
# the same or not.

a = 50
b= 599

count_a = 0
count_b = 0
while a != 0:
   a //= 10
   count_a += 1
print("Number of digits in a: " + str(count_a))

while b != 0:
   b //= 10
   count_b += 1
print("Number of digits in b: ", count_b)

if count_a == count_b:
   print("True")
else:
   print("False")
```

```
#second option
def sameLength(A, B):
   while (A > 0 and B > 0):
       A = A / 10;
       B = B / 10;
       # Both must be 0 now if
   # they had same lengths
   if (A == 0 and B == 0):
       return True;
   return False;


A = 20;
B = 10;

if (sameLength(A, B)):
   print("Yes");
else:
   print("No");


# 8) Write a function that takes in a number and determines if
# the digits contain two adjacent 5s.

def double_5(n):
   flag = 0
   while n > 0:
       n, d = n // 10, n % 10
       if d == 5 and flag == 1:
           return print("True")
       elif d == 5:
           flag = 1
       else:
           flag = 0
   return print("False")



double_5(50255)

# 9) Design a function that returns the number of unique digits in a positive
integer

def unique_digits(num):
   return len(set(str(num)))


print(unique_digits(1010101))
```