Exercise 5.5 - Group 6

1. a function that takes a number and returns a list of its all possible factors (e.g. for 10, these are [1, 2, 5, 10]). Name this function "factors"
2. a function that takes a number and returns "True" if the number is prime and "False" if the number is not prime. Name this function "is_prime"
3. a function that takes two numbers and returns their Highest Common Factor (e.g. for 12 and 18, it is 6). Name this function "hcf"

In [2]:
```
# EXERCISE 5.5

# FUNCTION GET USER INPUT
def get_user_input():
    print('Enter a number: ')
    #  GET USER INPUT
    user_input = input()
    # CHECK IF THE INPUT IS NUMERIC
    if user_input.isnumeric():
        # IF IT IS NUMERIC, MAKE SURE IT IS AN INTEGER
        user_input = int(user_input)
        # RETURN THE RESULT
        return user_input
    else:
        # PRINT NOT A NUMBER
        print("Not a Number")
    # RETURN THE RESULT
    return user_input


# CREATE A FACTOR FUNCTION
def find_factors(number):
    # CREATE A FACTOR LIST
    factors = []
    # FOR LOOP THAT INCREMENTS NUMBER FROM 1 (TO MAX OF NUMBER LENGTH)
    # THEN CHECK IF NUMBER MODULUS WHOLE_NUMBER
    # IF 0 THEN IT IS A FACTOR
    for whole_number in range(1, number + 1):
        # IF THE NUMBER MODULUS WHOLE_NUMBER == 0
        if number % whole_number == 0:
            # SAVE THIS TO A LIST FACTORS
            factors.append(whole_number)
    # RETURN THE RESULT
    return factors


# FUNCTION TO CHECK IF A NUMBER IS PRIME
def is_prime(number):
    # IF NUMBER IS LESS THEN 2 THEN NOT A PRIME
    if number < 2:
        return False
    # INCREMENT NUMBER FROM 2 TO MAX OF NUMBER LENGTH
    for i in range(2, number):
        # IF THIS NUMBER MODULUS 2 IS 0 THEN PRIME
        if not number % i:
```

```
            # RETURN FALSE
            return False
    # ELSE RETURN TRUE
    return True


# HIGHEST COMMON FACTOR TAKES IN TWO NUMBERS
def hcf(number_x, number_y):
    # THIS IF ELSE STATEMENT WILL
    # SWAP THE NUBERS IF NUMBER X IS SMALLER THEN NUMBER Y
    if number_x > number_y:
        x = number_y
    else:
        x = number_x
    # WHILE LOOP X IS LAREGR THEN 1
    # CHECK IF NUMBER X AND Y MODULUS X IS 0, IF NOT,
    # KEEP SUBTRACTING 1 FROM X UNTIL MODULUS X, Y = 0
    while x > 1:
        if number_x % x == 0 and number_y % x == 0:
            break
        x -= 1
    # RETURN X AS THE HIGHEST COMMON FACTOR
    return x

# TRY EXECUTE THE CODE AND REPORT ANY ERRORS
try:
    # GET THE USER INPUT
    user_input = get_user_input()
    # PASS THE USER INPUT INTO THE CHECK FACTORS FUNCTION
    check_factors = find_factors(user_input)
    # PASS THE USER INPUT INTO THE CHECK PRIME FUNCTION
    check_prime = is_prime(user_input)
    # GET THE HIGHEST COMMON FACTOR
    check_hcf = hcf(18, 12)

    # PRINT THE OUTPUTS FOR FACTORS
    print('Factors:', check_factors)
    # PRIME NUMBERS
    print('Is It A Prime Number', check_prime)
    # HIGHEST COMMON FACTORS
    print('Highest Common Factor', check_hcf)

# CATCH ANY ERRORS
except:
    print("An Error Occured")
```

```
Enter a number:

Factors: [1, 3, 9]
Is It A Prime Number False
Highest Common Factor 6
```

In [ ]: