EE4371 Assignment-1 by J Antonson (ee19b025)

Problem 1a

Write a Python function that takes a positive integer n, and returns the sum of the squares of all the positive integers smaller than n.

In [1]:

```
# sqr function to find out the sum of squares of integers less than m
def sqr(m):                  # function definition starts
    a=0
    for i in range(1,m):     # loop through all the values of all i with i<m & i>0 condition satisfied
        a= a + i**2          # take the square of i and add it to 'a'
    return a                 # returns the sum of the squares of all the positive integers smaller than m
```

In [2]:

```
number = int(input())
print("The sum of squares of positive integers less than {} is ".format(number) + str(sqr(number)))
```

```
5
The sum of squares of positive integers less than 5 is 30
```

Problem 1b

Write a Python function that takes a positive integer n, and returns the sum of the squares of all the odd positive integers smaller than n

In [3]:

```
# osqr function to find out the sum of squares of all ODD integers less than m

def osqr(m):                    # function definition starts
    a=0
    for i in range(1,m,2):      # loop through all the values of all ODD i with i<m & i>0 condition satisfied
        a = a + i**2            # take the square of i and add it to 'a'
    return a                    # returns the sum of the squares of all the odd positive integers smaller than m
```

In [4]:

```
number = int(input())
print("The sum of squares of positive ODD integers less than {} is ".format(number) + str(osqr(number)))
```

```
5
The sum of squares of positive ODD integers less than 5 is 10
```

Problem 2

What parameter values should be sent to the range constructor to produce a range with values:

- (a) 60,70,80
- (b) 4,2,0,-2,-4

```
range (60, 81, 10)
```

In [5]:

```
# 2.a
a=[]
for i in range(60,81,10):
    a.extend([i])
print(*a)                    #for checking the values
```

```
60 70 80
```

```
range(4, -5, -2)
```

In [6]:

```
# 2.b
b=[]
for i in range(4,-5,-2):
    b.extend([i])
print(*b)                  #for checking the values
```

```
4 2 0 -2 -4
```

Problem 3

Write a Python function that takes a sequence of integer values and determines if there is a distinct pair of numbers in the sequence whose product is odd

In [16]:

```
def ocheck(a):
    for i in a:
        for j in a:                 # nested for loop in order to access all the elements of the list
            if i!=j:                # check if the elements are distinct
                product = i*j       # if elements are distinct, take their product
                if product%2!=0:  # check if product is odd ----- if satisfied, return 'True'
                    print("YES- distinct pair of numbers in the sequence whose product is odd EXISTS")
                    return True
    print("NO- distinct pair of numbers in the sequence whose product is odd DOES NOT exist")
    return False                    # return 'False' otherwise (i.e "no" distinct pairs of numbers whose product is odd)
```

In [17]:

```
print(ocheck(list(map(int,input().split()))))   # input to the function ocheck is shredded to form a list (since the a
rgument for ocheck is a list)
```

```
2 4 3 6 7 4 8
YES- distinct pair of numbers in the sequence whose product is odd EXISTS
True
```

Problem 4

Write a Python function that counts the number of vowels in a given character string

In [9]:

```
def findvowel(a):                                             # creating a function to find the number of vowels in a
 given string
    a = a.lower()                                             # converting all the letters of the string into lower ca
se letters
    count=0                                                   # count is set to zero
    for i in a:                                               # iterating over the elements of the string
        if i=='a' or i=='e' or i=='i' or i=='o' or i=='u':    # comparing the elements of the string with vowels
            count+=1                                          # if the letter compared is a vowel, increment count by
 1
    return count                                              # retrun the value stored in count
```

In [11]:

```
print("number of vowels in the given string = " + str(findvowel(input())))
```

```
abcde fghij
number of vowels in the given string = 3
```

Problem 5

Write a Python program that takes as input three integers, “a”, “b” and “c”, from the console and determines if they can be used in the following arithmetic formulas:

- (i) “a+b=c”
- (ii) “a=b-c”
- (iii) “a*b=c”.

In [12]:

```
a= list(map(int,input().split()))                       # creating a list for ease

if a[0]+a[1] == a[2]:                                   # checking case (i)
    print("a+b=c or {}+{}={}".format(a[0],a[1],a[2]))
if a[0]+a[2] == a[1]:                                   # checking case (ii)
    print("a=b-c or {}={}-{}".format(a[0],a[1],a[2]))
if a[0]*a[1] == a[2]:                                   # checking case (iii)
    print("a*b=c or {}*{}={}".format(a[0],a[1],a[2]))
```

```
3 6 3
a=b-c or 3=6-3
```

Problem 6 (Project)

Design a program that can test the [Birthday problem (https://en.wikipedia.org/wiki/Birthday_problem)](https://en.wikipedia.org/wiki/Birthday_problem), by a series of experiments, on randomly generated birthdays which test this paradox for n=5,10,15,20,25,30...200.

In [13]:

```python
import random                                                                                    # impori
ng 'random' libarary to generate random numbers for testing the birthday problem
```

In [14]:

```
# Program for printing the test result values of Birthday problem

print("Enter the number of iterations to check on: ")
largetest= int(input())                                                                                       # gettin
g inputs from the keyboard (for the number of iterations to make) ---- LARGER the number you put, LONGER it will take
 for the code to  run

a=[]                                                                                                          # array
 for storing the tested value (which can be used for ploting later)


for i in range(5, 201, 5):                                                                                    # range
 constructor for iterating from 5 to 200 (inclusive) with a seperation of 5
    print("The number selected= "+ str(i))
    p=[]                                                                                                      # Binary
array used to store the success (True) and Failiure (False) of testcases to check the probability of each n

    for m in range(largetest):                                                                                # 'm' it
erating through [largetest]== number of iterations
        birthdays=[]                                                                                          # this a
rray is used to store the randomly generated numbers (birthdays)
        for k in range(i):                                                                                    # Loop f
or iterating n number of times (i==n)
            birthdays.extend([random.randrange(0,365)])                                                       # creati
ng random numbers from 0 to 365 and appending the elements to the list/array 'birthdays'

        if any(birthdays.count(element) > 1 for element in birthdays):                                        # if len
(birthdays) != len(set(birthdays)):
            p.append(True)                                                                                    # if the
re is atleast two people having the same birthday (i.e same number), then append 'True' to the array 'p'
        else:                                                                                                 # otherw
ise append 'False' to the array p
            p.append(False)

    a.append([float(sum(p))/float(largetest)*100])                                                            # append
the "probability of finding a 'True' (i.e atleast two people having the same birthday) from the array 'p'" to the arra
y 'a'
    print(str(a[int(i/5)-1][0])+ "%" +" on test \n")                                                          # print
 the probability
```

```
Enter the number of iterations to check on:
10000
The number selected= 5
2.7% on test

The number selected= 10
11.799999999999999% on test

The number selected= 15
25.990000000000002% on test

The number selected= 20
40.99% on test

The number selected= 25
55.720000000000006% on test

The number selected= 30
70.38% on test

The number selected= 35
81.08% on test

The number selected= 40
89.32% on test

The number selected= 45
94.07% on test

The number selected= 50
97.02% on test

The number selected= 55
98.72% on test

The number selected= 60
99.41% on test

The number selected= 65
99.72999999999999% on test
```

```
The number selected= 70
99.92999999999999% on test

The number selected= 75
99.98% on test

The number selected= 80
100.0% on test

The number selected= 85
99.99% on test

The number selected= 90
100.0% on test

The number selected= 95
100.0% on test

The number selected= 100
100.0% on test

The number selected= 105
100.0% on test

The number selected= 110
100.0% on test

The number selected= 115
100.0% on test

The number selected= 120
100.0% on test

The number selected= 125
100.0% on test

The number selected= 130
100.0% on test

The number selected= 135
100.0% on test
```

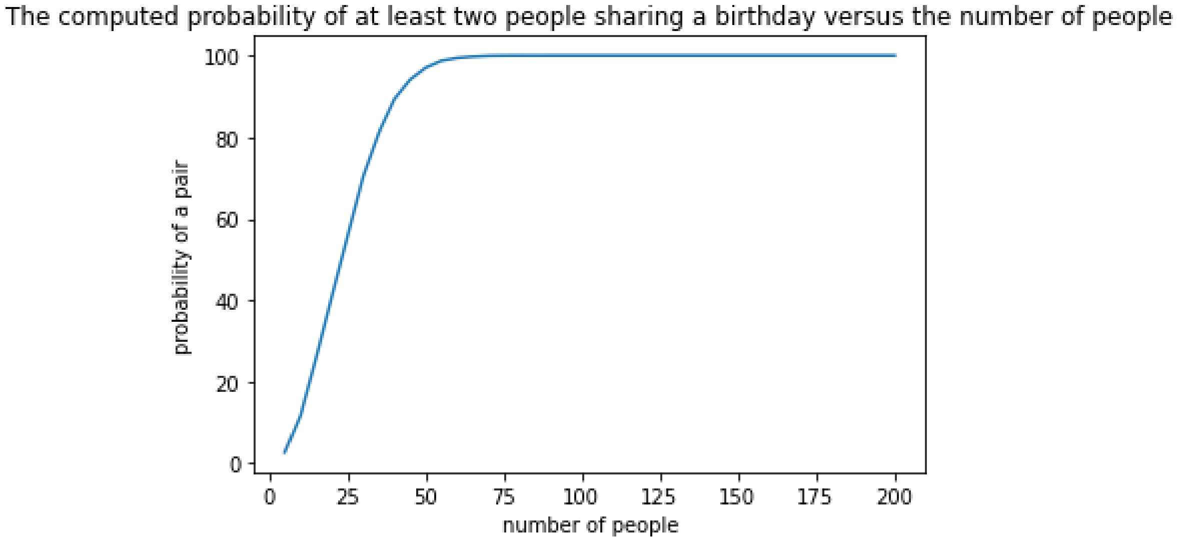
```
The number selected= 140
100.0% on test

The number selected= 145
100.0% on test

The number selected= 150
100.0% on test

The number selected= 155
100.0% on test

The number selected= 160
100.0% on test

The number selected= 165
100.0% on test

The number selected= 170
100.0% on test

The number selected= 175
100.0% on test

The number selected= 180
100.0% on test

The number selected= 185
100.0% on test

The number selected= 190
100.0% on test

The number selected= 195
100.0% on test

The number selected= 200
100.0% on test
```

In [15]:

```
# Code for plotting the tested data

import matplotlib.pyplot as plt                                                                         # impo
rting matplotlib libarary for plotting

x=[]
for i in range(5,201, 5):                                                                               # scal
ing the x-axis
  x.extend([i])

plt.plot(x, a)                                                                                          # plot
ting the graph of the computed probability of at least two people sharing a birthday versus the number of people

plt.title('The computed probability of at least two people sharing a birthday versus the number of people')
plt.xlabel('number of people')
plt.ylabel('probability of a pair')
# plt.xticks(np.arange(5, 201, 5))
```

Out[15]:

```
Text(0, 0.5, 'probability of a pair')
```



Addon codes for comparing the theoritical values with the Tested value.

Feel free to ignore the following code

```
def numerator(a, n):
    num=1
    b=a-n
    for i in range(b+1,a,1):
        num=num*i
    return float(num)
```

```
def denominator(a, n):
    den=1
    for i in range(n-1):
        den= den*a
    return float(den)
```

```
def probability(n, numberofdays):
    if n<=120:                                               # ease of computation & Precise answers for values less than or equal to 120
        num= numerator( numberofdays ,n)
        den= denominator( numberofdays ,n)
        probab= 1- num/den
        return probab*100
    else:                                                    # Not so precice, but can be computed for larger numbers easily
        probab_inv=1
        for i in range(n):
            probab_inv= probab_inv*(1 - float(n)/float(numberofdays))
        return probab_inv
```

```
# Program for printing the test result values of Birthday problem

print("Enter the number of iterations to check on: ")
largetest= int(input())                                                                                     #getting inputs
from the keyboard (for the number of iterations to make) ---- LARGER the number you put, LONGER it will take for the code t
o  run

a=[]                                                                                                        #array for stor
ing the tested value (which can be used for ploting later)


################################ Please remove the following comments from line 8 to line 11 if you want to check and compa
re the tested values with the theoritical values

b=[] #array for storing the theoritical value
c=[] #array for storing the error (for later analysis puroses)

#-------------------------------



for i in range(5, 201, 5):                                                                                    # range const
ructor for iterating from 5 to 200 (inclusive) with a seperation of 5
    print("The number selected= "+ str(i))
    p=[]                                                                                                      # Binary arra
y used to store the success (True) and Failiure (False) of testcases to check the probability of each n

    for m in range(largetest):                                                                                # 'm' iterati
ng through [largetest]== number of iterations
        birthdays=[]                                                                                          # this array 
 is used to store the randomly generated numbers (birthdays)
        for k in range(i):                                                                                    # loop for it
erating n number of times (i==n)
```

```
            birthdays.extend([random.randrange(0,365)])                                                       # creating ra
ndom numbers from 0 to 365 and appending the elements to the list/array 'birthdays'

        if any(birthdays.count(element) > 1 for element in birthdays):                                        # if len(birt
hdays) != len(set(birthdays)):
            p.append(True)                                                                                    # if there is
atleast two people having the same birthday (i.e same number), then append 'True' to the array 'p'
        else:                                                                                                 # otherwise a
ppend 'False' to the array p
            p.append(False)

    a.append([float(sum(p))/float(largetest)*100])                                                            # append the
 "probability of finding a 'True' (i.e atleast two people having the same birthday) from the array 'p'" to the array 'a'
    print(str(a[int(i/5)-1][0])+ "%" +" on test \n")                                                          # print the p
robability




############################## Please remove the following comments from line 36 to line 49 if want to compare and the test
ed value with the theoritical value

    if i<=120:
        b.append([probability(i,365)])
        print(str(b[int(i/5)-1][0])+ "%" + "  on theory \n")

        c.append([100- probability(i,365)-float(sum(p))/float(largetest)*100])
        print("Theory - test= " + str(c[int(i/5)-1][0]) + "% \n\n")
    else:
        b.append([100- probability(i,365)])
        print("(100 - "+ str(b[int(i/5)-1][0])+ ") %" + "  on theory \n")

        c.append([100 - probability(i,365) -float(sum(p))/float(largetest)*100])
        print("Theory - test= " + str(c[int(i/5)-1][0]) + "% \n\n")

#------------------------------
```

```
import matplotlib.pyplot as plt

x=[]
for i in range(5,201, 5):
  x.extend([i])

plt.plot(x, a)
plt.plot(x, b)
plt.plot(x, c)
```