

מבוא לתכנות מדעי פיתון
תרגיל בית 9
שותפים: נימר נסייר 322626896
פאטמה נעמה 212100580

שאלה 1:

Link: https://py3.codeskulptor.org/#user306_LHIMUS5twB_0.py

```
#Question 1
#3 functions to write
def FindDivisors(n):
    #list for put all the divisors in
    list_d=[]
    for i in range(1,n+1):
        # if our number is able to be divided by i
        #then add him to the list
        if n%i==0:
            list_d.append(i)
    return list_d
def IsPrime(n):
    #checking if n is prime
    #start from 2 because 2 is prime
    i=2
    #i^2 is lower or equal n
    while(i*i<=n):
        #if n divided by i with no remainder then its NOT
        if n%i==0:
            return False
        i=i+1
    return True
import math
# A function to print all prime factors of
# a given number n
def Decompose(n):
    list_c=[]
    # add the number of two's that divide n to the list
    while n%2==0:
        list_c.append(2)
        n/=2
    # n must be odd at this point
    # so a skip of 2 ( i = i + 2) can be used
    for i in range(3,int(math.sqrt(n))+1,2):
        # while i divides n , add i and divide n
        while n%i==0:
            list_c.append(int(i))
            n/=i
    # Condition if n is a prime
    if n>2:
        list_c.append(int(n))
    return list_c
d={}
for n in range(2,201):
    if IsPrime(n)==False:
        d[n]=(FindDivisors(n),Decompose(n))
    print(n,d[n])
```

```

172 ([1, 2, 4, 8, 16, 32, 64, 128], [2, 2, 4, 8])
174 ([1, 2, 3, 6, 29, 58, 87, 174], [2, 3, 8])
175 ([1, 5, 7, 25, 35, 175], [5, 17])
176 ([1, 2, 4, 8, 11, 16, 22, 44, 88, 176], [2, 2, 2, 2, 11])
177 ([1, 3, 59, 177], [3, 17])
178 ([1, 2, 89, 178], [2, 89])
180 ([1, 2, 3, 4, 5, 6, 9, 10, 12, 15, 18, 20, 30, 36, 45, 60, 90,
180], [2, 2, 3, 4])
182 ([1, 2, 7, 13, 14, 26, 91, 182], [2, 7, 9])
183 ([1, 3, 61, 183], [3, 18])
184 ([1, 2, 4, 8, 23, 46, 92, 184], [2, 2, 2, 23])
185 ([1, 5, 37, 185], [5, 18])
186 ([1, 2, 3, 6, 31, 62, 93, 186], [2, 3, 9])
187 ([1, 11, 17, 187], [11, 18])
188 ([1, 2, 4, 47, 94, 188], [2, 2, 47])
189 ([1, 3, 7, 9, 21, 27, 63, 189], [3, 18])
190 ([1, 2, 5, 10, 19, 38, 95, 190], [2, 5, 9])
192 ([1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 64, 96, 192], [2, 2, 2, 2,
2, 2, 3])
194 ([1, 2, 97, 194], [2, 97])
195 ([1, 3, 5, 13, 15, 39, 65, 195], [3, 19])
196 ([1, 2, 4, 7, 14, 28, 49, 98, 196], [2, 2, 7, 4])
198 ([1, 2, 3, 6, 9, 11, 18, 22, 33, 66, 99, 198], [2, 3, 9])
200 ([1, 2, 4, 5, 8, 10, 20, 25, 40, 50, 100, 200], [2, 2, 2, 5, 2])

```

```

4 ([1, 2, 4], [2, 2])
6 ([1, 2, 3, 6], [2, 3])
8 ([1, 2, 4, 8], [2, 2, 2])
9 ([1, 3, 9], [3])
10 ([1, 2, 5, 10], [2, 5])
12 ([1, 2, 3, 4, 6, 12], [2, 2, 3])
14 ([1, 2, 7, 14], [2, 7])
15 ([1, 3, 5, 15], [3])
16 ([1, 2, 4, 8, 16], [2, 2, 2, 2])
18 ([1, 2, 3, 6, 9, 18], [2, 3])
20 ([1, 2, 4, 5, 10, 20], [2, 2, 5])
21 ([1, 3, 7, 21], [3, 2])
22 ([1, 2, 11, 22], [2, 11])
24 ([1, 2, 3, 4, 6, 8, 12, 24], [2, 2, 2, 3])
25 ([1, 5, 25], [5, 2])
26 ([1, 2, 13, 26], [2, 13])
27 ([1, 3, 9, 27], [3, 2])
28 ([1, 2, 4, 7, 14, 28], [2, 2, 7])
30 ([1, 2, 3, 5, 6, 10, 15, 30], [2, 3])
32 ([1, 2, 4, 8, 16, 32], [2, 2, 2, 2, 2])
33 ([1, 3, 11, 33], [3, 3])
34 ([1, 2, 17, 34], [2, 17])

```

שאלה 2:

Link: https://py3.codeskulptor.org/#user306_LHIMUS5twB_1.py

בשאלה זאת שימוש המילון מהיר יותר מ list

```
#Question 2
#import random,time
import random
import time
#first ge the list
dlist=[random.random() for i in range(1000000)]
#dictionary
d1={random.random():1 for i in range(1000000)}
#x as requested
x=0.3
#1000 times for the list to run
for i in range(1000):
    time_start=time.time()
    if x in dlist:
        continue
end_time=time.time()
#our time after not finiding x
ourT=end_time-time_start
print("Time using list",ourT)
#1000 times for dictionary
for i in range(1000):
    #start time
    time_start2=time.time()
    if x in d1:
        continue
end_time2=time.time()
#end time and get our time in the dictionary
ourT=end_time2-time_start2
print("Time using dict",ourT)
## as we can see dictionary uses no time(0)
#which makes it alot faster than list which takes a little bit
#but it will make a difference in big projects
```

```
Time using list 0.010833024978637695
Time using dict 0.0
```

Link: https://py3.codeskulptor.org/#user306_LHIMUS5twB_2.py

```
import math
import matplotlib.pyplot as plt
#get our inputs
n=int(input("Please enter n: "))
p=float(input("Please enter p: "))
boolean=str(input("Please enter True or False: "))
#our help function to get benomi
def prob(X,n,p):
    #BENOMI TRICK rule
    a=math.factorial(n)
    b=math.factorial(X)
    a_b=math.factorial(n-X)
    div= a /(b*(a_b))
    benomi=div * p**X *(1-p)**(n-X)
    #returning "UNTIL" 3 after decimal
    x=float("{0:.3f}".format(benomi))
    return x
def BinomialPMF(n,p,plot=False):
    # two empty list to fill the numbers and their probabilities
    list_i=[]
    list_p=[]
    #our dictionary with its values, probabilities and their keys
    list_vp={'Values':list_i,'Probability':list_p}
    #fill the numbers and probabilities in lists
    for i in range(n+1):
        list_i.append(i)
    for j in range(n+1):
        #use our prob func to find the probability
        list_p.append(prob(j, n, p))
    #if plot is true then we draw the bar
    if plot=='True':
        plt.bar(list_i,list_p)
        plt.show()
    return list_vp
w=BinomialPMF(n,p,boolean)
print(w)
```

Please enter n: 5

Please enter p: 0.4

Please enter True or False: True

{'Values': [0, 1, 2, 3, 4, 5], 'Probability': [0.078, 0.259, 0.346, 0.23, 0.077, 0.01]}

