

מבוא לתכנות מדעי פיתון  
תרגיל בית 10  
שותפים: נימר נסייר 322626896  
פאטמה נעמה 212100580

שאלה 1:

Link: [https://py3.codeskulptor.org/#user306\\_kzR12YWoLs\\_0.py](https://py3.codeskulptor.org/#user306_kzR12YWoLs_0.py)

```
#Question 1
def RecFib(n):
    if n<=1:
        return n
    else:
        #fibonaci is the sum of the last two numbers
        #so we keep going until we reach n
        #and add n and n+1 as recursive works
        return (RecFib(n-1) + RecFib(n-2))
n=int(input("Please enter a positive integer: "))
print(RecFib(n))
```

```
Please enter a positive integer: 2
1
```

```
In [13]: runfile('/Users/nemernser/Question 1.py',
wdir='/Users/nemernser')
```

```
Please enter a positive integer: 4
3
```

```
In [14]: runfile('/Users/nemernser/Question 1.py',
wdir='/Users/nemernser')
```

```
Please enter a positive integer: 6
8
```

Link: [https://py3.codeskulptor.org/#user306\\_kzR12YWoLs\\_1.py](https://py3.codeskulptor.org/#user306_kzR12YWoLs_1.py)

```
#Question 2
def RecPalindrome(W):
    if len(W)<1:
        return 1
    else:
        #palidrome is if the word is the same backwards
        #check if first element and last is equal
        if W[0]==W[-1]:
            #if yes they we check the middle(without first and the last)
            return RecPalindrome(W[1:-1])
        else:
            return 0
W=str(input("Please enter a string: "))
if RecPalindrome(W):
    print(W,"is a palindrome")
else:
    print(W,"is NOT a palindrome")
```

```
Please enter a string: aba
aba is a palindrome
```

```
In [16]: runfile('/Users/nemernser/Q
wdir='/Users/nemernser')
```

```
Please enter a string: abccba
abccba is a palindrome
```

```
In [17]: runfile('/Users/nemernser/Q
wdir='/Users/nemernser')
```

```
Please enter a string: aaad
aaad is NOT a palindrome
```

שאלה 3:

Link: [https://py3.codeskulptor.org/#user306\\_kzR12YWoLs\\_2.py](https://py3.codeskulptor.org/#user306_kzR12YWoLs_2.py)

```
#Question 3
def RecCat(n):
    #when to stop recursive
    if n<1:
        return 1
    else:
        #katlan is sum of katlan i * katlan n-i-1
        #C0 is 1 as the question
        kat=0
        for i in range(n):
            # get the sum of i * n-i-1
            #because n is n>=1 we wait until it is smaller than 1
            kat+=RecCat(i)*RecCat(n-i-1)
        return kat
katlan=int(input("Please enter a number: "))
print(RecCat(katlan))
```

```
Please enter a number: 3
5
```

```
In [21]: runfile('/Users/nemernser/
Users/nemernser')
```

```
Please enter a number: 5
42
```

```
In [22]: runfile('/Users/nemernser/
Users/nemernser')
```

```
Please enter a number: 6
132
```

Link: [https://py3.codeskulptor.org/#user306\\_kzR12YWoLs\\_3.py](https://py3.codeskulptor.org/#user306_kzR12YWoLs_3.py)

```
import matplotlib.pyplot as plt
#Question 4
f=open("/Users/nemernser/.spyder-py3/Grades.txt","r")
#find the least 2 homeworks
#list to get student names
student_list=[]
#list to fill with the students homework
list_a=[]
#list to fill with the final marks
students_final=[]
#our dictionary
ourmilon={}
sum1=0
for line in f:
    x=line.split()
    #10 studnents
    #8 homeworks remove 2 lowest
    #2 exams
    #fill student list with the student names
    student_list.append(str(x[0]))
    #go from 1(without the names) to 9 not included(all the homeworks)
    for i in range(1,9):
        list_a.append(x[i])
    #get the 2 lowest marks for this student
    min1=sorted(list_a)[0]
    min2=sorted(list_a)[1]
    #remove these 2 homeworks
    for element in list_a:
        if min1 in list_a:
            list_a.remove(min1)
        if min2 in list_a:
            list_a.remove(min2)
    sum1=0
    #get the average for the homework for this student
    for j in list_a:
        sum1+=int(j)
    avg=sum1/len(list_a)
    #make the formula for the final grade
    #x[9] is the middle exam, x[10] is the final exam
    students_final.append(avg*0.2+int(x[9])*0.3+int(x[10])*0.5)
    #round our list
    students_rounded=[round(num) for num in students_final]
    #clear the list for the other student to calculate the marks
    list_a=[]
    #because our marks will be lists we get them one by one
    #and insert the mark from our list to the dictionary in the place the student is
    for element in students_rounded:
        ourmilon[str(x[0])]=element
    #clear the lists for other students to calculate
    students_final=[]
    students_rounded=[]
print(ourmilon)
#to plot a dictionary we need to make the keys and the values
keys=list(ourmilon.keys())
values=list(ourmilon.values())
plt.xticks(fontsize=5)
plt.xlabel("Students names")
plt.ylabel("Students Final mark")
plt.bar(keys,values)
```

```
In [23]: runfile('/Users/nemernser/Question 4.py',  
wdir='/Users/nemernser')  
{'Student1': 72, 'Student2': 79, 'Student3': 87,  
'Student4': 74, 'Student5': 91, 'Student6': 59,  
'Student7': 93, 'Student8': 75, 'Student9': 64,  
'Student10': 58}
```

