

# Homework No.1. Autor: John Robert Garnica Prada

August 6, 2019

## 1 Exercise 1

Generate odd numbers. Write a program that generates all odd numbers from 1 to n. Set n in the beginning of the program and use a while loop to compute the numbers. (Make sure that if n is an even number, the largest generated odd number is n-1.)

```
[77]: print('Please enter a integer number N:')
N = int(input())
i = 0
while i <= N:
    if i%2 != 0:
        print(i)
    i = i + 1;
```

Please enter a integer number N:

5  
1  
3  
5

## 2 Exercise 2

Store odd numbers in a list. Modify the program from the previous exercise to store the generated odd numbers in a list. Start with an empty list and use a while loop where you in each pass of the loop append a new element to the list. Finally, print the list elements to the screen.

```
[78]: print('Please enter a integer number N:')
N = int(input())
i = 0
oddN=[]
while i <= N:
    if i%2 != 0:
        oddN.append(i)
    i = i + 1;
oddN
```

Please enter a integer number N:

5

[78]: [1, 3, 5]

### 3 Exercise 3

Generate odd numbers by a list comprehension. Solve Exercise 2 using a list comprehension (with for and range).

```
[79]: print('Please enter a integer number N:')
N = int(input())
i = 0
oddN=[]
for i in range(N + 1):
    if i%2 != 0:
        oddN.append(i)
    i = i + 1;
oddN
```

Please enter a integer number N:  
5

[79]: [1, 3, 5]

### 4 Exercise 4

Make a table of values from a formula. Write a program that prints a nicely formatted table of t and y(t) values, where

$$y(t) = (v_0 t) - (1/2)g(t^2)$$

Use n uniformly spaced t values throughout the interval [0, 2v<sub>0</sub>/g]. Set v<sub>0</sub>=1 and n = 11.

```
[89]: vo = 1
t0 = 0
n = 11
g= 9.81
columnas = 2
Table = [[0 for x in range(columnas)] for y in range(n)]
for i in range(n):
    t0 = t0 + ((2*vo)/g)/n
    y0 = (vo*t0) - (1/2)*g*(t0*t0)
    Table[i][0] = t0
    Table[i][1] = y0
Table
```

```
[89]: [[0.01853396348809193, 0.016849057716447208],
[0.03706792697618386, 0.030328303889604974],
[0.055601890464275786, 0.0404377385194733],
[0.07413585395236771, 0.04717736160605218],
[0.09266981744045964, 0.05054717314934162],
```

```
[0.11120378092855157, 0.05054717314934162],
[0.12973774441664349, 0.04717736160605218],
[0.14827170790473543, 0.040437738519473285],
[0.16680567139282737, 0.030328303889604946],
[0.1853396348809193, 0.016849057716447152],
[0.20387359836901126, -5.551115123125783e-17]]
```

## 5 Exercise 5

Store values from a formula in lists. Modify the program from Exercise 4 so that the  $t$  and  $y$  values are stored in two lists  $t$  and  $y$ . Thereafter, transverse the lists with a for loop and write out a nicely formatted table of  $t$  and  $y$  values (using either a zip or range construction). Set  $v_0 = 10$  and  $n = 81$ .

```
[143]: vo = 10
t0 = 0
n = 81
g= 9.81
columnas = 2
t = []
y = []
for i in range(n):
    t0 = t0 + ((2*vo)/g)/n
    y0 = (vo*t0) - (1/2)*g*(t0*t0)
    t.append(t0)
    y.append(y0)
Table = [[0 for x in range(columnas)] for y in range(len(t))]
for i in range(len(t)):
    Table[i][0] = t[i]
    Table[i][1] = y[i]
Table
```

```
[143]: [[0.025169580045556937, 0.24858844489438953],
[0.050339160091113874, 0.4909621786664193],
[0.0755087401366708, 0.7271212013160893],
[0.10067832018222775, 0.9570655128433997],
[0.1258479002277847, 1.1807951132483503],
[0.15101748027334164, 1.398310002530941],
[0.17618706031889858, 1.6096101806911722],
[0.20135664036445552, 1.8146956477290435],
[0.22652622041001247, 2.013566403644555],
[0.2516958004555694, 2.206222448437707],
[0.2768653805011263, 2.392663782108499],
[0.3020349605466832, 2.572890404656931],
[0.32720454059224013, 2.7469023160830037],
[0.35237412063779705, 2.9146995163867166],
[0.37754370068335397, 3.0762820055680695],
[0.4027132807289109, 3.2316497836270623],
```

[0.4278828607744678, 3.380802850563697],  
 [0.4530524408200247, 3.52374120637797],  
 [0.47822202086558163, 3.660464851069885],  
 [0.5033916009111385, 3.7909737846394385],  
 [0.5285611809566955, 3.915268007086634],  
 [0.5537307610022524, 4.033347518411468],  
 [0.5789003410478093, 4.145212318613944],  
 [0.6040699210933662, 4.250862407694059],  
 [0.6292395011389231, 4.350297785651816],  
 [0.65440908118448, 4.443518452487211],  
 [0.679578661230037, 4.5305244082002485],  
 [0.7047482412755939, 4.611315652790924],  
 [0.7299178213211508, 4.685892186259242],  
 [0.7550874013667077, 4.754254008605198],  
 [0.7802569814122646, 4.816401119828797],  
 [0.8054265614578215, 4.872333519930033],  
 [0.8305961415033785, 4.922051208908911],  
 [0.8557657215489354, 4.965554186765429],  
 [0.8809353015944923, 5.002842453499589],  
 [0.9061048816400492, 5.033916009111387],  
 [0.9312744616856061, 5.058774853600827],  
 [0.956444041731163, 5.077418986967905],  
 [0.98161362177672, 5.0898484092126255],  
 [1.0067832018222769, 5.096063120334984],  
 [1.0319527818678338, 5.0960631203349855],  
 [1.0571223619133907, 5.0898484092126255],  
 [1.0822919419589476, 5.077418986967907],  
 [1.1074615220045045, 5.058774853600826],  
 [1.1326311020500615, 5.033916009111389],  
 [1.1578006820956184, 5.00284245349959],  
 [1.1829702621411753, 4.965554186765432],  
 [1.2081398421867322, 4.922051208908914],  
 [1.2333094222322891, 4.872333519930036],  
 [1.258479002277846, 4.816401119828799],  
 [1.283648582323403, 4.754254008605201],  
 [1.3088181623689599, 4.685892186259245],  
 [1.3339877424145168, 4.611315652790928],  
 [1.3591573224600737, 4.530524408200252],  
 [1.3843269025056306, 4.443518452487217],  
 [1.4094964825511875, 4.35029778565182],  
 [1.4346660625967445, 4.250862407694065],  
 [1.4598356426423014, 4.1452123186139485],  
 [1.4850052226878583, 4.033347518411476],  
 [1.5101748027334152, 3.91526800708664],  
 [1.5353443827789721, 3.790973784639446],  
 [1.560513962824529, 3.660464851069891],  
 [1.585683542870086, 3.523741206377979],

```
[1.6108531229156429, 3.3808028505637058],
[1.6360227029611998, 3.2316497836270717],
[1.6611922830067567, 3.0762820055680784],
[1.6863618630523136, 2.914699516386724],
[1.7115314430978705, 2.7469023160830144],
[1.7367010231434274, 2.572890404656942],
[1.7618706031889844, 2.3926637821085084],
[1.7870401832345413, 2.2062224484377158],
[1.8122097632800982, 2.0135664036445675],
[1.8373793433256551, 1.8146956477290566],
[1.862548923371212, 1.6096101806911811],
[1.887718503416769, 1.3983100025309518],
[1.9128880834623259, 1.1807951132483652],
[1.9380576635078828, 0.957065512843414],
[1.9632272435534397, 0.7271212013161019],
[1.9883968235989966, 0.49096217866643244],
[2.0135664036445537, 0.24858844489439846],
[2.0387359836901107, 1.4210854715202004e-14]]
```

## 6 Exercise 6

Work with a list. Set a variable `primes` to a list containing the numbers 2, 3, 5, 7, 11, and 13. Write out each list element in a for loop. Assign 17 to a variable `p` and add `p` to the end of the list. Print out the whole new list.

```
[108]: primes = [2,3,5,7,11,13]
primes2 = [0,0,0,0,0,0]
for i in range(len(primes)):
    primes2[i] = primes[i]
    i = i + 1
p=17
primes2.append(p)
primes2
```

```
[108]: [2, 3, 5, 7, 11, 13, 17]
```

## 7 Exercise 7

Simulate operations on lists by hand. You are given the following program:

```
a = [1,3,5,7,11] b = [13,17] c = a+b print c b[0] = -1 d = [e+1 for e in a] print d d.append(b[0] +
1) d.append(b[-1] + 1) print d[-2:]
```

Go through each statement and explain what is printed by the program.

```
[128]: a = [1,3,5,7,11]
```

Generate a list `a`, with elements 1, 3, 5, 7 and 11.

```
[129]: b = [13,17]
```

Generate a list b, with elements 13 and 17.

```
[130]: c = a + b
```

Calculate the summary of list a and b.

```
[132]: print c
```

```
File "<ipython-input-132-1b1d2b33e8c2>", line 1
print c
    ^
```

SyntaxError: Missing parentheses in call to 'print'. Did you mean print(c)?

Program generates an Error, cause the variable printed must be in parenthesis.

```
[133]: b[0] = -1
```

Modifies position zero of list b.

```
[135]: d = [e+1 for e in a]
```

Create a new list from a list elements, the new values will be the value of list plus 1.

```
[ ]: print d
```

Program generates an Error, cause the variable printed must be in parenthesis

```
[137]: d.append(b[0] + 1)
```

Add a element at the end of list d, the value will be the first element of list plus 1.

```
[ ]: d.append(b[-1] + 1)
```

Add a element at the end of list d, the value will be the last element of list plus 1.

```
[138]: print d[-2:]
```

```
File "<ipython-input-138-63cc55da71d2>", line 1
print d[-2:]
    ^
```

SyntaxError: Missing parentheses in call to 'print'. Did you mean print(d[-2:↵])?

Program generates an Error, cause the variable printed must be in parenthesis.