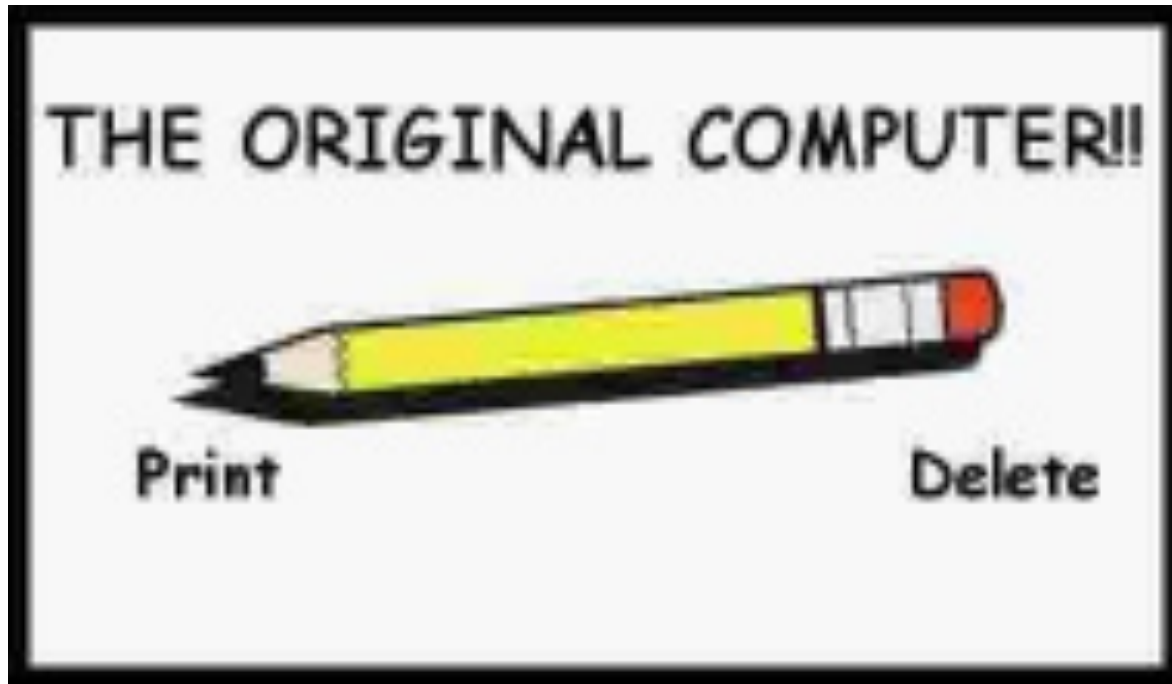


ME1 Computing



Provide feedback (anonymously) at:

www.menti.com

with code **44 88 7**

Nested loops

The function $y(x) = \frac{1}{1-x}$ can be represented by the series expansion:

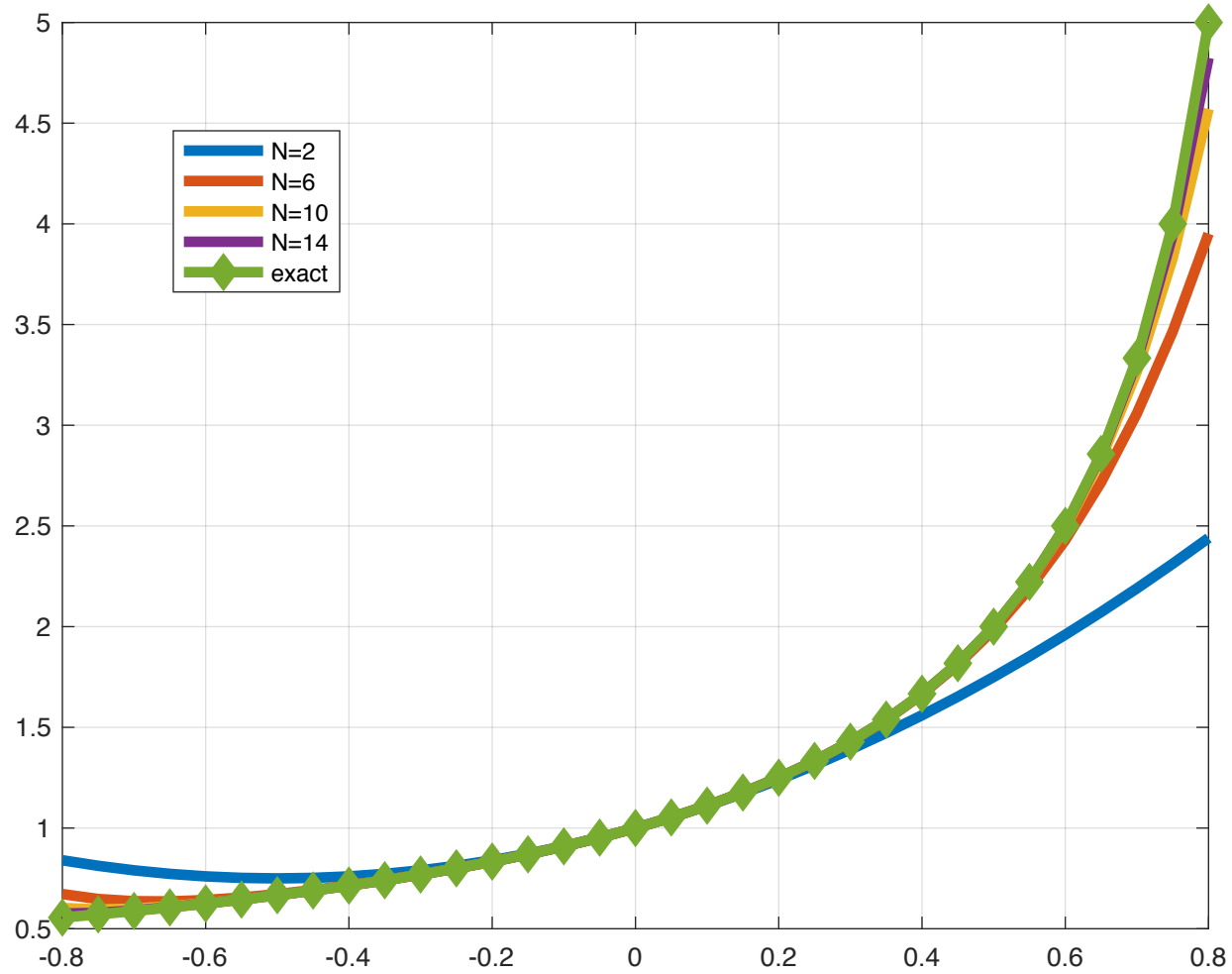
$$y(x) = \frac{1}{1-x} = \sum_{i=0}^{N \rightarrow \infty} x^i = 1 + x + x^2 + x^3 + x^4 + \dots$$

in the interval $-1 < x < 1$ only.

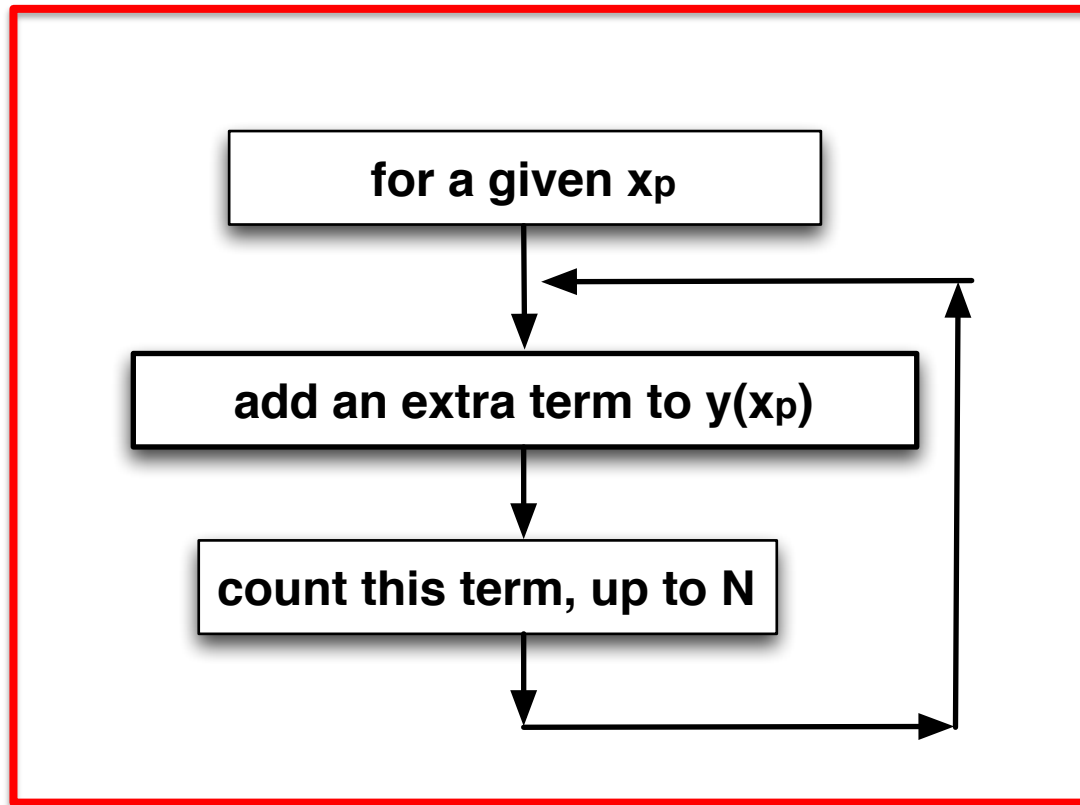
Write a script to evaluate the function $y(x)$ in the range $x = [-0.8 \ 0.8]$ with step 0.01 , for values of $N = 2, 6, 10, 14$.

Plot, on the same graph, $y(x)$ vs x in the specified range $x = [-0.8 \ 0.8]$, for each value of N .

Nested loops



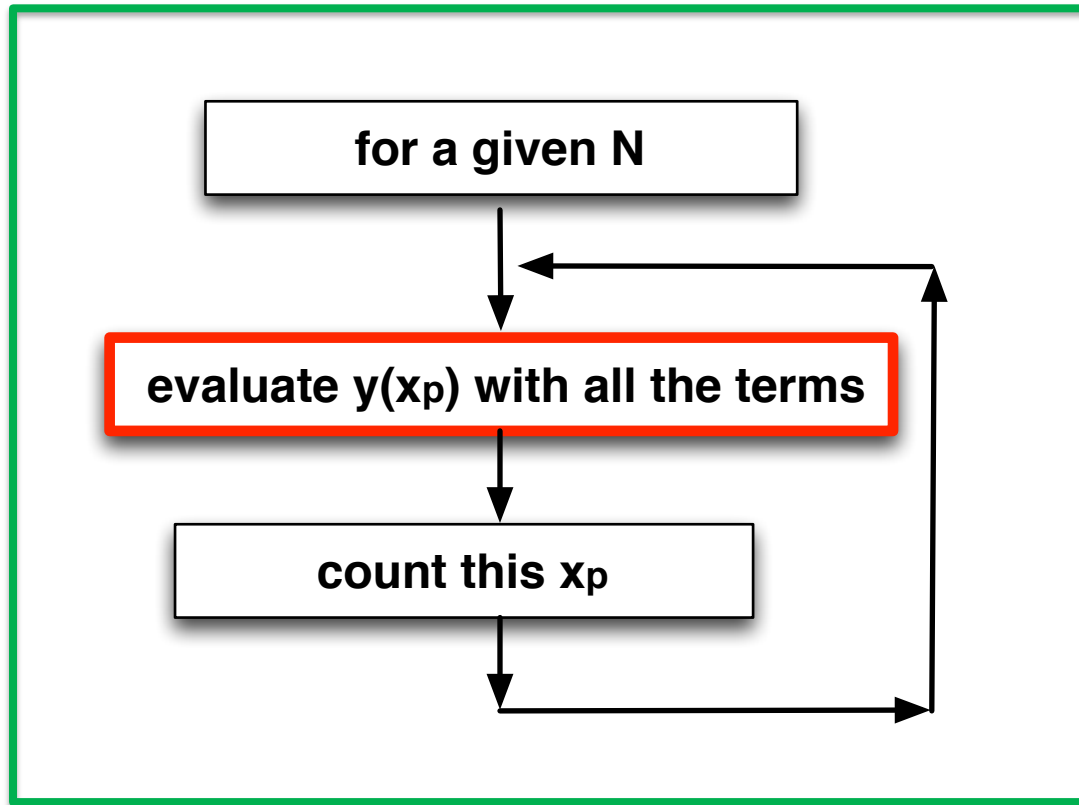
Nested loops



$$y(x_p) = \sum_{i=0}^N x_p^i$$

For all point $x = [-0.8 \ 0.8]$

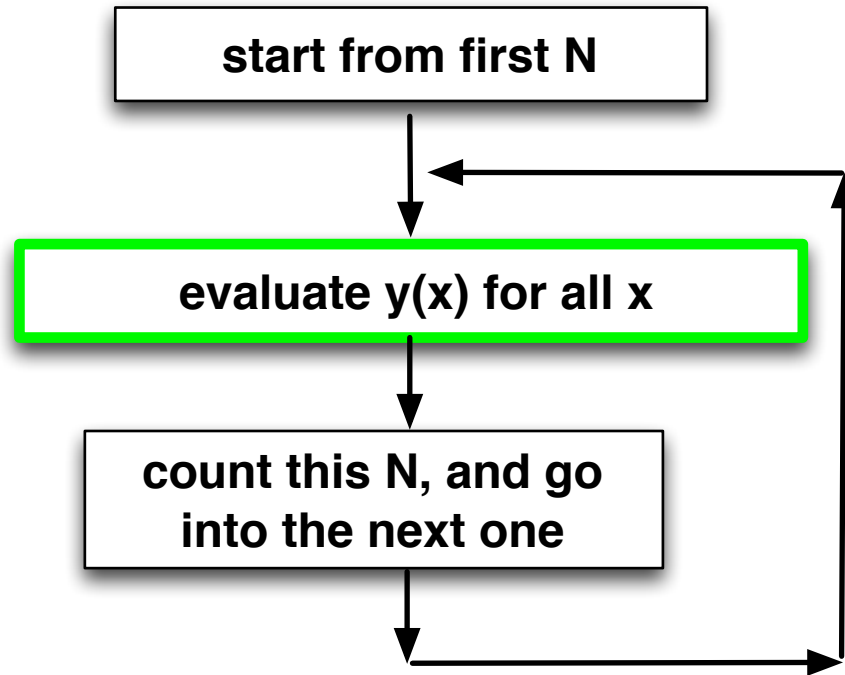
Nested loops



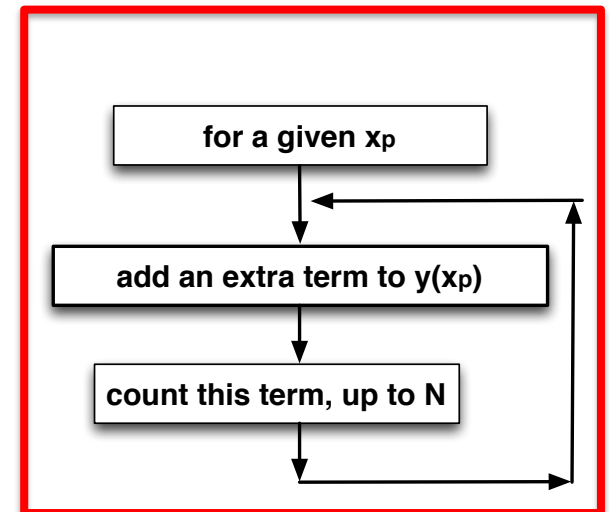
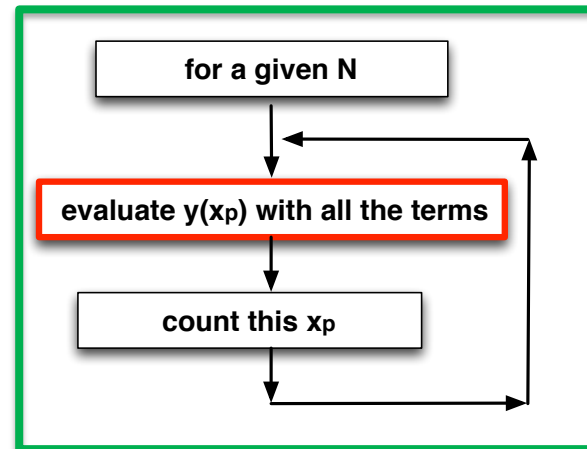
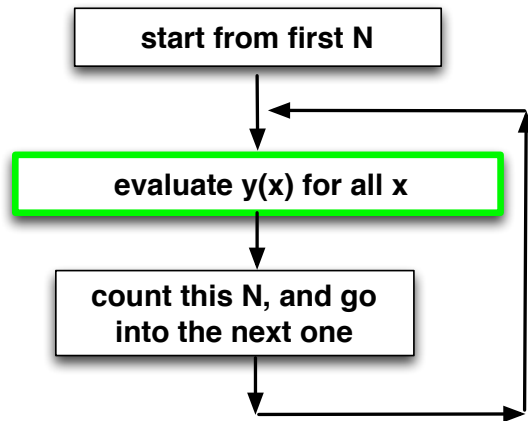
$$y(x) = \sum_{i=0}^N x^i$$

For N = 2, 6, 10, 14

Nested loops



Nested loops



The Search algorithm

Given a list of numbers search if a given number is present in the list

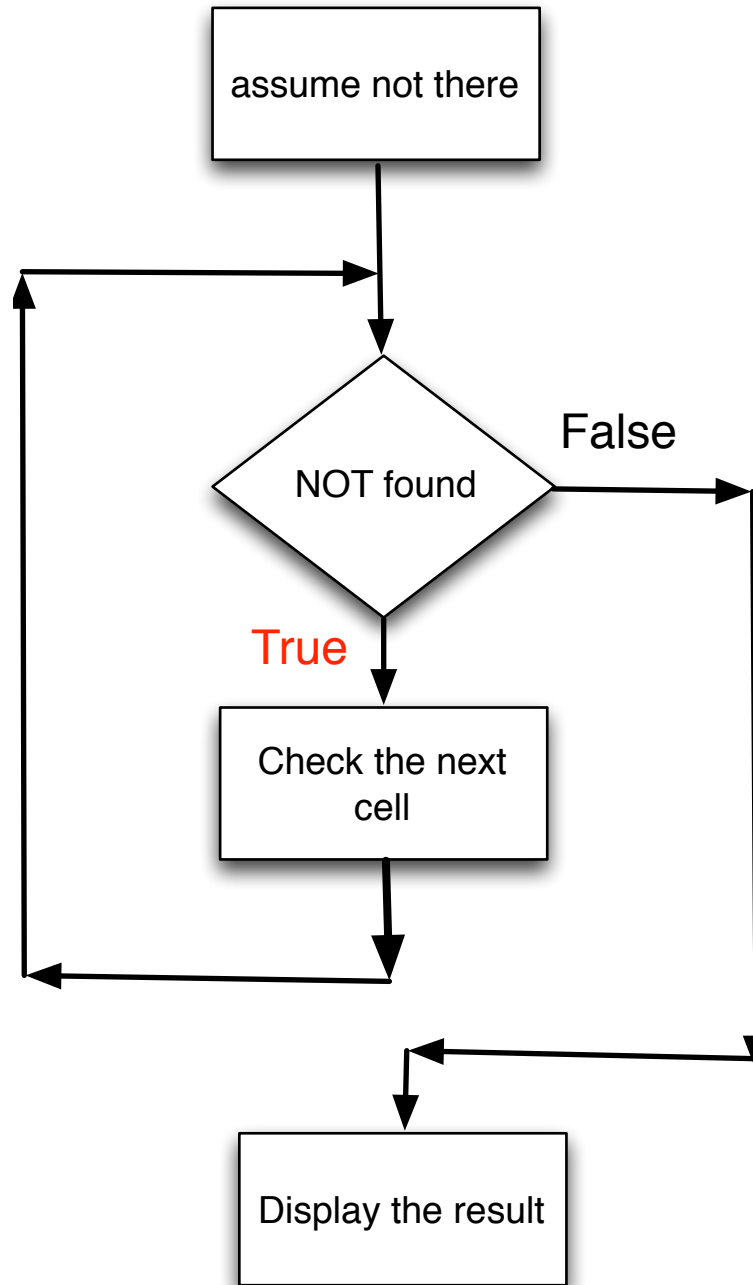
Bingo	13	24	5	8	33	44	10	45	2	25
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

Search if 8 is present

```
find = 8
found = False
for i in Bingo:
    if i == find:
        found = True
#
print(found)
```


Conditional loops

Loop repeated as long
as the condition is true



Conditional loops: WHILE construct

```
while condition:  
    % do it while the condition is TRUE
```

Conditional loops: WHILE construct

Bingo

13	24	5	8	33	44	10	45	2	25
(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

```
find = 33 # number to be sought
found = False
count = 0
while (not found):
    if Bingo[count] == find:
        found = True
    else:
        count = count + 1
#
print(found)
```

WHILE loop vs FOR loop

We need to keep the count ourselves

```
find = 8
found = False
for i in Bingo:
    if i == find:
        found = True
#
print(found)
```

FOR LOOP

```
find = 33
found = False
count = 0
while (not found):
    if Bingo[count] == find:
        found = True
    else:
        count = count + 1
#
print(found)
```

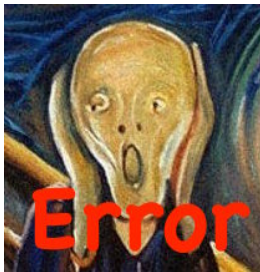
WHILE LOOP

WHILE construct: common pitfalls

We need to keep the count ourselves

```
find = 33
found = False
while (not found):
    if Bingo[count] == find:
        found = True
#
print(found)
```

```
find = 33
found = False
count = 0
while (not found):
    if Bingo[count] == find:
        found = True
    else:
        count = count + 1
#
print(found)
```



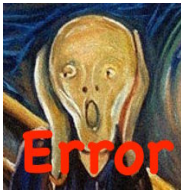
CORRECT

WHILE construct: common pitfalls

Make sure you do not over exceed list dimensions

Bingo	13	24	5	8	33	44	10	45	2	25
	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

```
find = 18
found = False
count = 0
while (not found):
    if Bingo[count] == find:
        found = True
    else:
        count = count + 1
```



INCOMPLETE

```
find = 18
N = len(Bingo)
found = False
count = 0
while (not found) and count < N:
    if Bingo[count] == find:
        found = True
    else:
        count = count + 1
```

COMPLETE

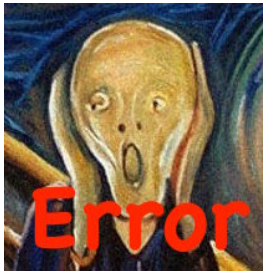
WHILE construct: common pitfalls

Make sure you that the condition will be true at some point

```
i = 1;  
while i < 10:  
    print('Hello')
```

```
i = 1  
while i > 10:  
    i = i + 1;
```

ENDLESS LOOP



Will never happen



WHILE construct: a useful application

Validate correctness of input data

Input a positive number

```
n = input('Gimme a positive')
while n<0:
    print('Not positive. Try again')
    n = input('Gimme a positive')
end
```

Rock, paper, scissor

```
hand = input('Gimme R, P or S')
while hand != 'R' & hand != 'P' & hand != 'S':
    print('Idiot. Try again')
    hand = input('Gimme r, p or s')
```


Loops: counted or conditional?

Do we know how many iterations/repetitions are needed beforehand?

yes

no

Use counted loops

Use conditional loops

```
for i in Range:  
    % do something
```

```
while condition:  
    % do something
```