Lesson 4

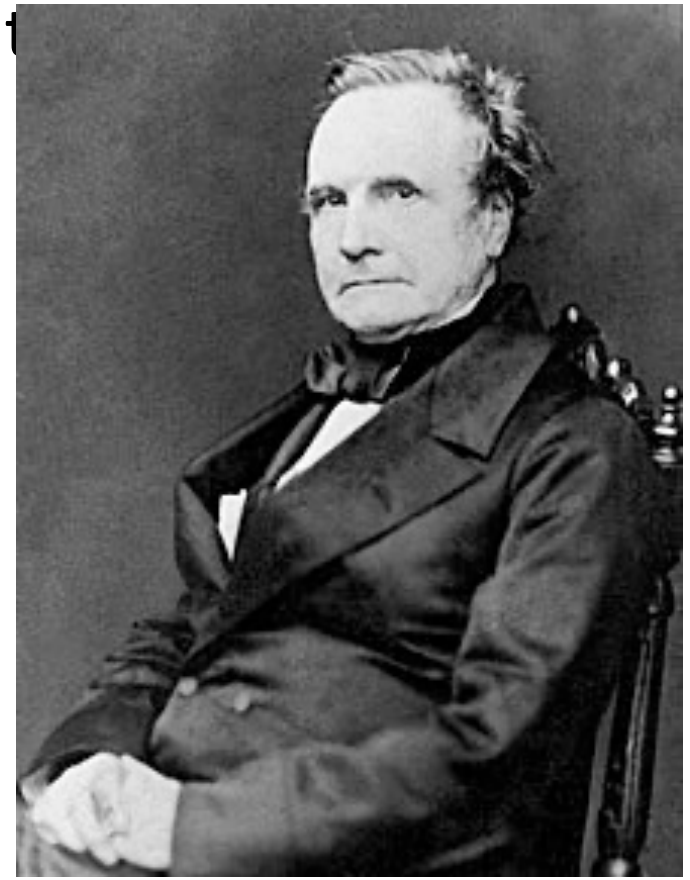
min max average storage . Remove letter .

function in function . Calendar of Month problem .

Reverse Cipher

Reminder: Who is Charles Babbage ?

- Interesting facts or stories ?
- What is his most significant work or contribution to t
  - Contribute in forum



Example – min, max, average & storage

- A group of students had a test (0 – 100) marks.
- We want to know the following:
  - Lowest mark
  - Highest mark
  - Average mark
  - Keep the score of all students

Exa

```
total = 0
minimum = 100 #assuming the maximum score is 100
maximum = 0   #assuming the minimum score is 0
storage = []

loop = int(input('Enter the number of input: '))

for i in range(loop):
    score = int(input('Enter the score: '))
    storage.append(score)
    total += score
    if score > maximum:
        maximum = score
    if score < minimum:
        minimum = score

average = total / loop

print('The minimum score is',minimum)
print('The maximum score is',maximum)
print('The average score is',average)
```

What if you want to stop the program by keying in an empty score. (another way to implement)

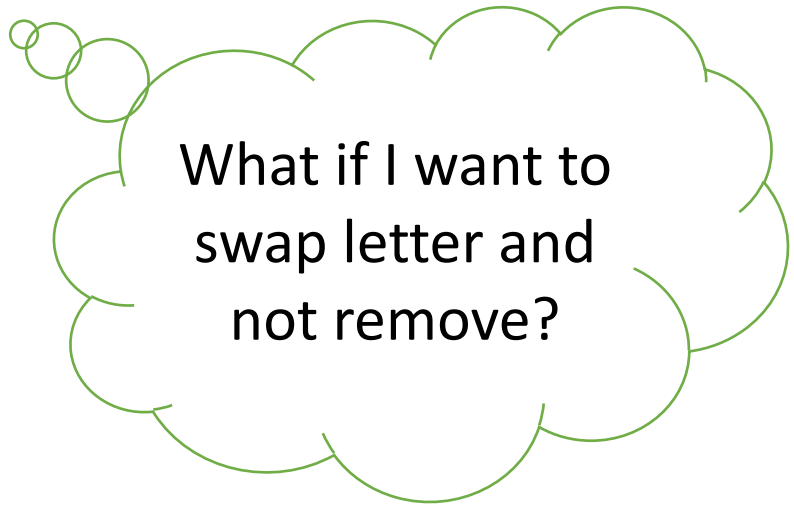
Example - Remove letter in string

Write a function `remove_letter(msg,letter)` that takes a string `msg` and a single character `letter`. This function returns a string, which is `msg` with all occurrences of `letter` removed. This function is case sensitive, so `'A'` is not the same as `'a'`. A sample run is shown below.

```
>>> message = "a pack of pickled peppers"
>>> remove_letter(message,'p')
"a ack of ickled eers"
```

Example - Remove letter in string

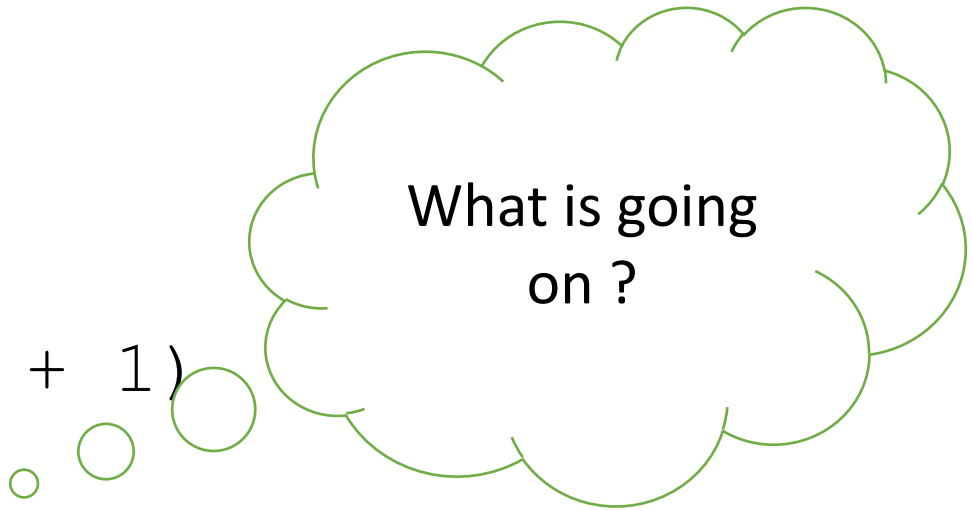
```
def remove_letter(msg, letter):
    result = ''
    for char in msg:
        if char != letter:
            result += char
    return result
```



let’s think . . . and discuss . . .

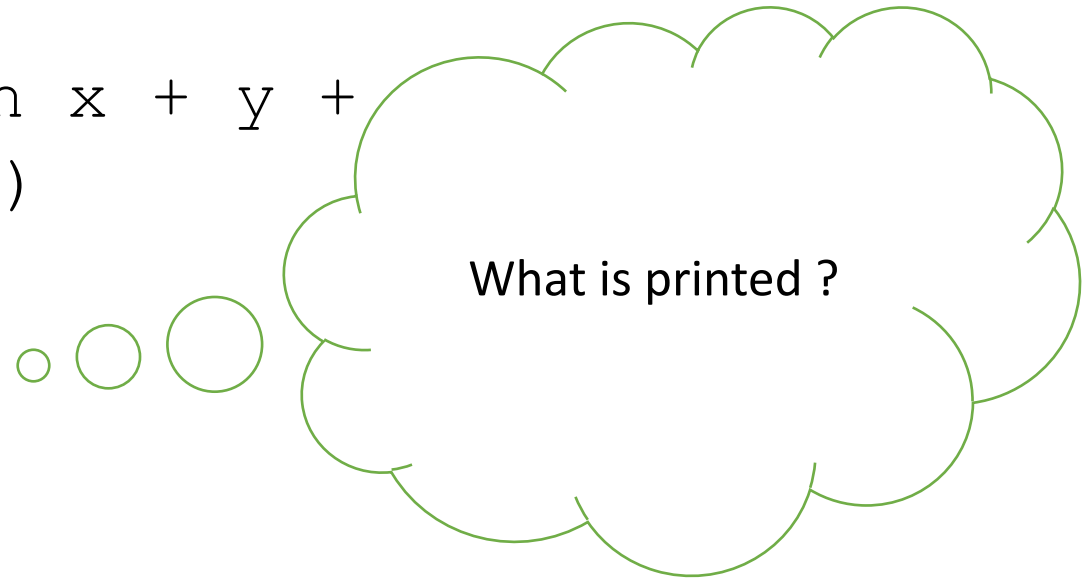
```
def f(g):
    return g(2)
def square(x):
    return x**2
def mystery(x):
    return x * (x + 1)
>>>f(square)

>>>f(mystery)
```



function in function

```
x = 5
y = 10
z = 15
def f(x):
    def g(y):
        def h(z):
            return x + y +
        return h(y)
    return g(x)
print(f(6))
```



continue and break

- The `continue` keyword causes Python to skip the remaining commands in the loop and go straight to deciding whether the loop should run again by testing the condition after the `while` keyword.
- The `break` keyword, on the other hand, causes Python to immediately skip the remaining commands and exit the loop completely.

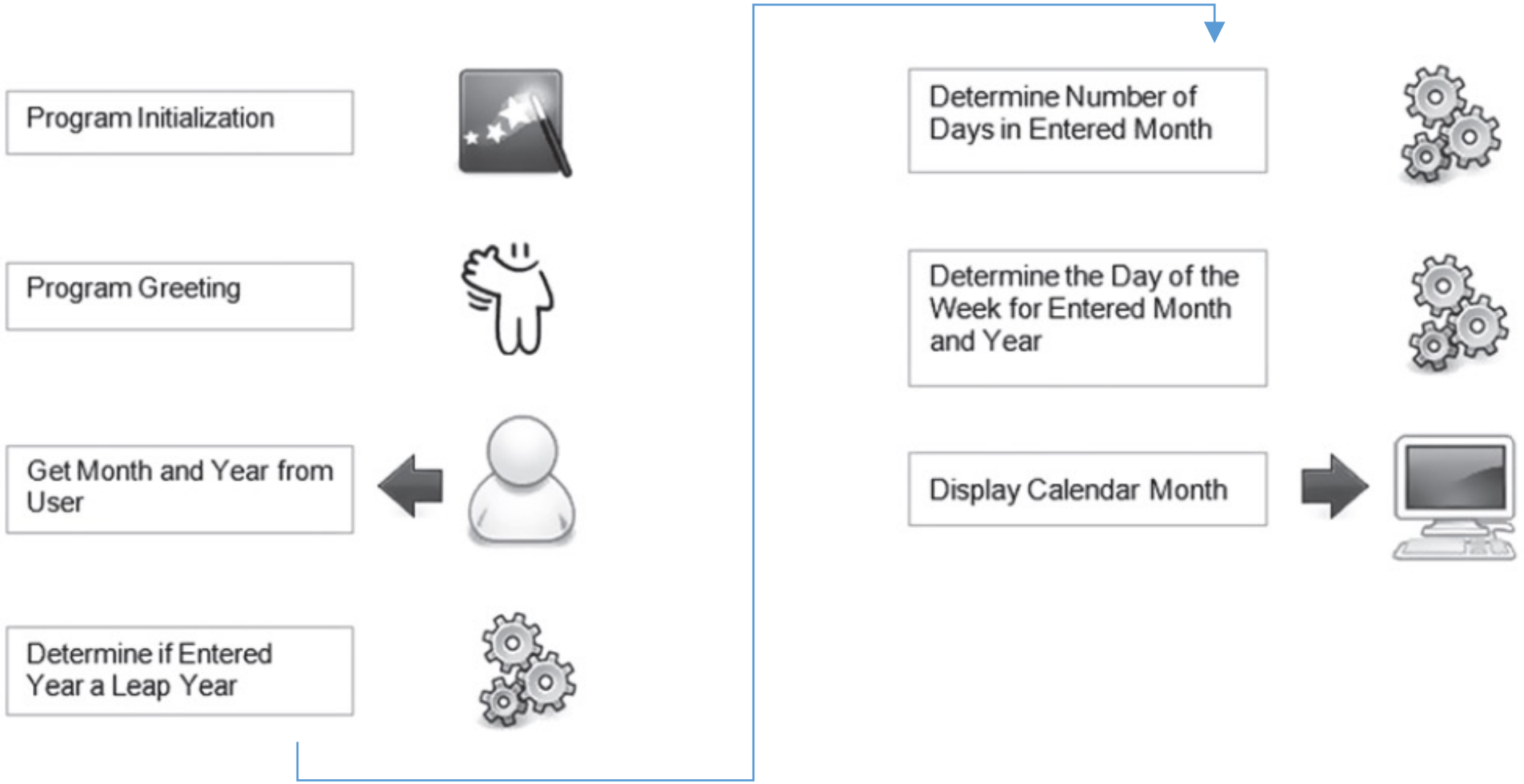
```
1   # Variable declarations
2   # words : list
3   # index : int
4
5   words = ["Code", "Computing", "Future", "Cow", "Thinking"]
6   index = 0
7   while index < len(words):
8       if words[index][0] != 'C':
9           index = index + 1
10          continue
11      print("C is for " + words[index])
12      index = index + 1
```

```
============== RESTART: C:/Examples/firstletter_continue.py ==============
C is for Code
C is for Computing
C is for Cow
>>>
```

```
1   # Variable declarations
2   # words : list
3   # index : int
4
5   words = ["Code", "Computing", "Future", "Cow", "Thinking"]
6   index = 0
7   while index < len(words):
8       if words[index][0] != 'C':
9           break
10      print("C is for " + words[index])
11      index = index + 1
```

```
================ RESTART: C:/Examples/firstletter_break.py ================
C is for Code
C is for Computing
>>>
```

Calendar of month Problem



Calendar of month Problem

```
This program will display a calendar month between 1800 and 2099

Enter month 1-12 (-1 to quit): 1
Enter year (yyyy): 1800

  January   1800
                      1   2   3
   4   5   6   7   8   9  10
  11  12  13  14  15  16  17
  18  19  20  21  22  23  24
  25  26  27  28  29  30  31

Enter month (1-12): -1
>>>
```

Cipher

A cipher is an algorithm which can be used to encode a message so that unintended recipients of the message are not able to read it.

- Plain Text

The plain text message is the text which is readable and can be understood by all users. The plain text is the message which undergoes cryptography.

- Cipher Text

Cipher text is the message obtained after applying cryptography on plain text.

- Encryption

The process of converting plain text to cipher text is called encryption. It is also called as encoding.

Reverse Cipher

Algorithm of Reverse Cipher

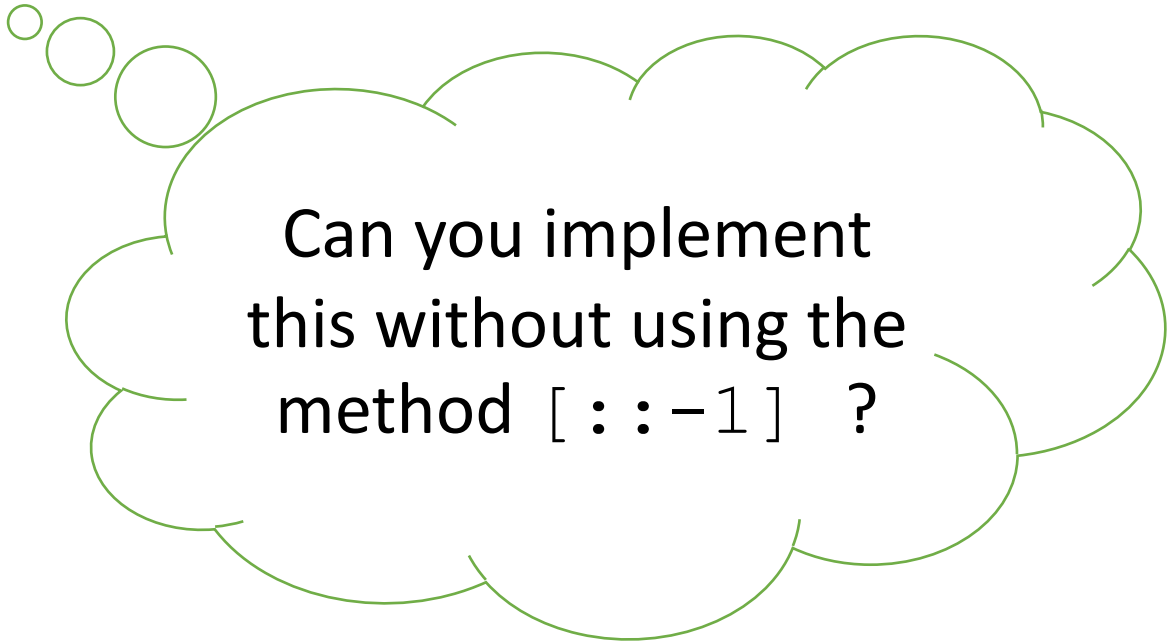
The algorithm of reverse cipher holds the following features –

- Reverse Cipher uses a pattern of reversing the string of plain text to convert as cipher text.
- The process of encryption and decryption is same.
- To decrypt cipher text, the user simply needs to reverse the cipher text to get the plain text.

The major drawback of reverse cipher is that it is very weak. A hacker can easily break the cipher text to get the original message. Hence, reverse cipher is not considered as good option to maintain secure communication channel.

Reverse Cipher

```
def reverse_s(string):
    return string[::-1]
```



Caesar Cipher

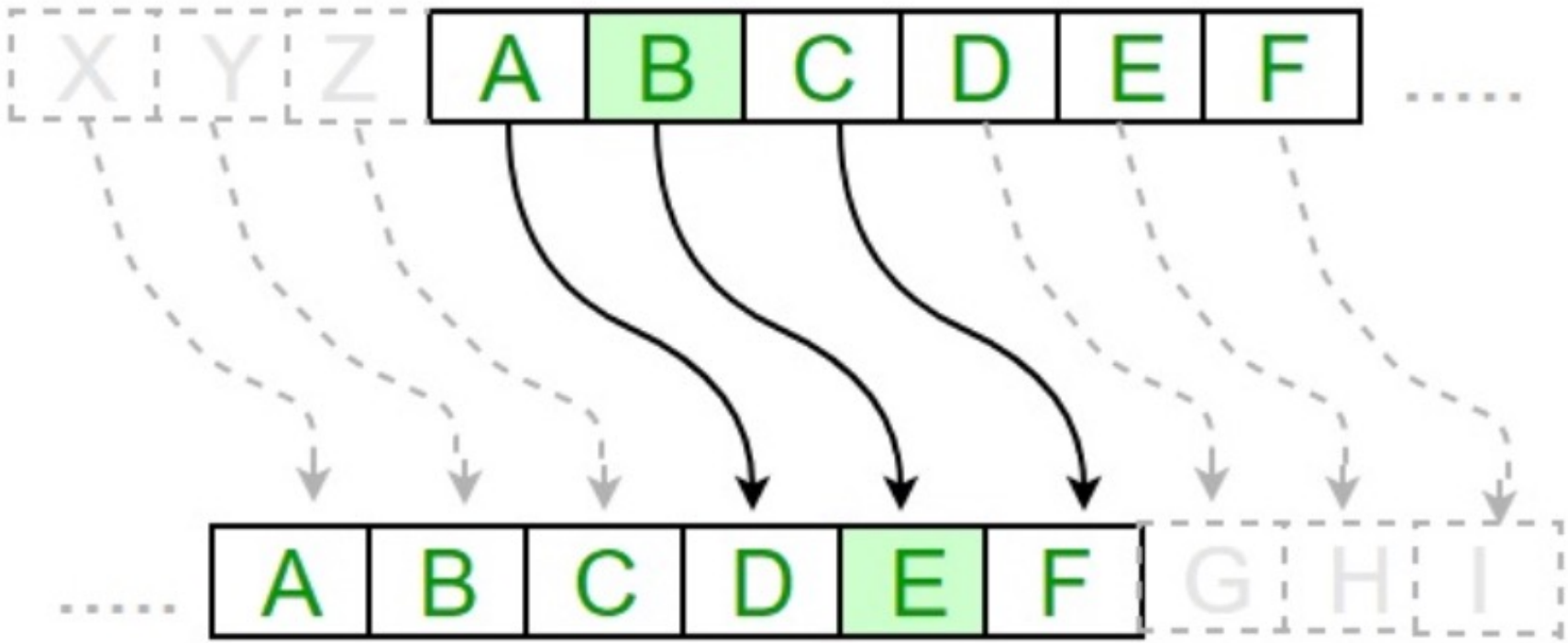
The Caesar Shift Cipher is a type of substitution cipher where each letter in the original plaintext message is replaced by a letter some fixed number of positions down the alphabet.

It is named after Julius Caesar, who used it in his private correspondence.

The table below shows an example where the letters have been shifted to the right by 3.

Note that the letters “wrap-around” at the end.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c


X Y Z
A B C D E F
A B C D E F G H I

shift in one character

```
def shift(char):
    '''
    Cycles alphabets, preserving case:
    a -> b -> c -> ... -> y -> z -> a
    Does nothing to other characters.
    '''
    if char.isalpha():
        return chr(ord(char)+1)
    else:
        return char
```

string.isalpha() returns True if is an alphabet, False otherwise

Something is missing !

Work to do . . . . . .

- 4A – Iteration
- Revision 1 (Fundamentals)
- Programming Assignment 4
- Calendar of Month Problem
- Cipher:
  - Reverse cipher (alternate implementation)
  - Shift one character