# TASK-1
## Caesar Cipher Program

**Implement Caesar Cipher:**

Create a Python program that can encrypt and decrypt text using the Caesar Cipher algorithm. Allow users to input a message and a shift value to perform encryption and decryption.

### What is **Caesar Cipher**?

This is an encryption algorithm that works by shifting all the letters in a message by some fixed numbers.

| Z | A | B | C | D |
|---|---|---|---|---|

| | A | B | C | D | E |
|---|---|---|---|---|---|
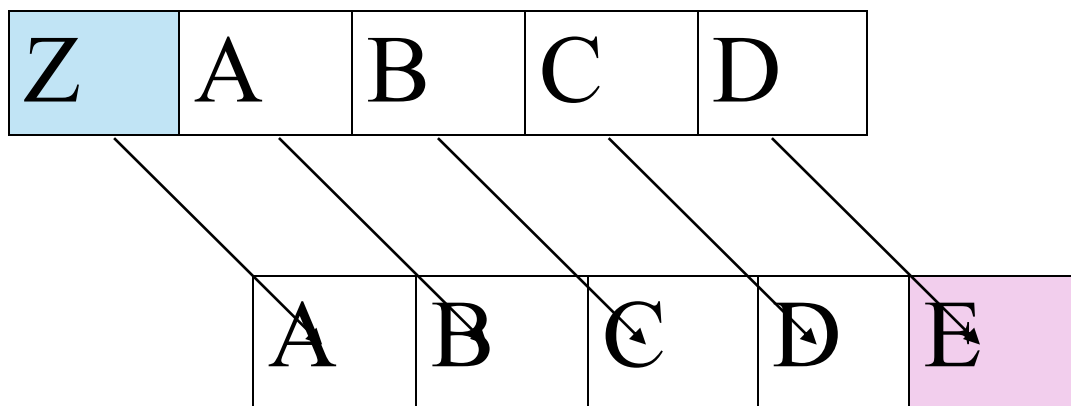
By applying the clipher with a shift f1

## "CAB" ⟶ "DBC"

Actually, we have binary numbers representation of each alphabet (ASCII codes)
Let us assume, A=65 & B=66   this is universally accepted standard

## A ⟶ 65 ⟶ 72 ⟶ H

Convert & Shifting

Here, we use python to convert each letter in our message to ASCII code number, then by shifting it by adding the shift number & we'll convert that result number back into a letter

So, by doing to all letters in our message then we'll end up with an encrypted message.

ASCII table which we use to see the binary number of the letter & identifying.

# ASCII Table

| Dec | Hex | Oct | Char | Dec | Hex | Oct | Char | Dec | Hex | Oct | Char | Dec | Hex | Oct | Char |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | | 32 | 20 | 40 | [space] | 64 | 40 | 100 | @ | 96 | 60 | 140 | ` |
| 1 | 1 | 1 | | 33 | 21 | 41 | ! | 65 | 41 | 101 | A | 97 | 61 | 141 | a |
| 2 | 2 | 2 | | 34 | 22 | 42 | " | 66 | 42 | 102 | B | 98 | 62 | 142 | b |
| 3 | 3 | 3 | | 35 | 23 | 43 | # | 67 | 43 | 103 | C | 99 | 63 | 143 | c |
| 4 | 4 | 4 | | 36 | 24 | 44 | $ | 68 | 44 | 104 | D | 100 | 64 | 144 | d |
| 5 | 5 | 5 | | 37 | 25 | 45 | % | 69 | 45 | 105 | E | 101 | 65 | 145 | e |
| 6 | 6 | 6 | | 38 | 26 | 46 | & | 70 | 46 | 106 | F | 102 | 66 | 146 | f |
| 7 | 7 | 7 | | 39 | 27 | 47 | ' | 71 | 47 | 107 | G | 103 | 67 | 147 | g |
| 8 | 8 | 10 | | 40 | 28 | 50 | ( | 72 | 48 | 110 | H | 104 | 68 | 150 | h |
| 9 | 9 | 11 | | 41 | 29 | 51 | ) | 73 | 49 | 111 | I | 105 | 69 | 151 | i |
| 10 | A | 12 | | 42 | 2A | 52 | * | 74 | 4A | 112 | J | 106 | 6A | 152 | j |
| 11 | B | 13 | | 43 | 2B | 53 | + | 75 | 4B | 113 | K | 107 | 6B | 153 | k |
| 12 | C | 14 | | 44 | 2C | 54 | , | 76 | 4C | 114 | L | 108 | 6C | 154 | l |
| 13 | D | 15 | | 45 | 2D | 55 | - | 77 | 4D | 115 | M | 109 | 6D | 155 | m |
| 14 | E | 16 | | 46 | 2E | 56 | . | 78 | 4E | 116 | N | 110 | 6E | 156 | n |
| 15 | F | 17 | | 47 | 2F | 57 | / | 79 | 4F | 117 | O | 111 | 6F | 157 | o |
| 16 | 10 | 20 | | 48 | 30 | 60 | 0 | 80 | 50 | 120 | P | 112 | 70 | 160 | p |
| 17 | 11 | 21 | | 49 | 31 | 61 | 1 | 81 | 51 | 121 | Q | 113 | 71 | 161 | q |
| 18 | 12 | 22 | | 50 | 32 | 62 | 2 | 82 | 52 | 122 | R | 114 | 72 | 162 | r |
| 19 | 13 | 23 | | 51 | 33 | 63 | 3 | 83 | 53 | 123 | S | 115 | 73 | 163 | s |
| 20 | 14 | 24 | | 52 | 34 | 64 | 4 | 84 | 54 | 124 | T | 116 | 74 | 164 | t |
| 21 | 15 | 25 | | 53 | 35 | 65 | 5 | 85 | 55 | 125 | U | 117 | 75 | 165 | u |
| 22 | 16 | 26 | | 54 | 36 | 66 | 6 | 86 | 56 | 126 | V | 118 | 76 | 166 | v |
| 23 | 17 | 27 | | 55 | 37 | 67 | 7 | 87 | 57 | 127 | W | 119 | 77 | 167 | w |
| 24 | 18 | 30 | | 56 | 38 | 70 | 8 | 88 | 58 | 130 | X | 120 | 78 | 170 | x |
| 25 | 19 | 31 | | 57 | 39 | 71 | 9 | 89 | 59 | 131 | Y | 121 | 79 | 171 | y |
| 26 | 1A | 32 | | 58 | 3A | 72 | : | 90 | 5A | 132 | Z | 122 | 7A | 172 | z |
| 27 | 1B | 33 | | 59 | 3B | 73 | ; | 91 | 5B | 133 | [ | 123 | 7B | 173 | { |
| 28 | 1C | 34 | | 60 | 3C | 74 | < | 92 | 5C | 134 | \ | 124 | 7C | 174 | | |
| 29 | 1D | 35 | | 61 | 3D | 75 | = | 93 | 5D | 135 | ] | 125 | 7D | 175 | } |
| 30 | 1E | 36 | | 62 | 3E | 76 | > | 94 | 5E | 136 | ^ | 126 | 7E | 176 | ~ |
| 31 | 1F | 37 | | 63 | 3F | 77 | ? | 95 | 5F | 137 | _ | 127 | 7F | 177 | |

We are going through different Phases in the project

# Phase1:

Printing original char code & Caesar cipher shifted char code

We print statement to the letters,
We get ASCII number or Unicode numbers or code character & now,
        We have char code,
We can just add the number shift to that char code & we'll get new char code (encrypted)

# Phase 2:

We can see whole transform (shift) of the char

# Phase 3:

Finally, to get full result of our encryption we need to add all of this char back together to form descriptive message
For this,
        We just create, an empty string

We can see here, adding the character & finally have the encrypted string as our output.

# Phase 4:

Print, out of loop (no longer loop needed)
In the output we see special char & spaces with a semi column

Now we going to fix this
Now loop ignore any char that isn't a part of the alphabet, we print them as it is.

# Phase 5:

We add "char. isalpha"
This statement will return a Boolean (it is a value tells either T/F)

Output: Now we can see it skips space between characters

# Phase 6:

In this phase,
We goanna add original char

# Phase 7:

We can see funky things start happening to some of my other char

So, this shift actually applying to all the lower-case letters
So, lower case char are carrying trouble

We can just convert the whole message to uppercase & then perform.
Another python function called "upper"

## Phase 8:
Adding "last_char_code = 90"
Because in ASCII code, Z is 90 (last alphabet)
& add, char_range = 26

Output: now little bit readble

## Phase 9:
I just went to encrypt a lot of diff messages at once (or) even if I want to to reverse the shift & use it to decrypt the messages
To do that I need to organise this code into a logical unit that I can re-use & that can has an interface that's easy to understand
        By putting this code into a function

"def aesar_shift()"
It tells what arguments it accepts like, input to a function
Arguments we have message & shift

Output: there is no output because, we didn't use the function yet!

## Phase 10:
Use the function & now also we should get the same result.

## Phase 11:
I can use it to decode a Caesar cipher
Here, we adding first char code = 65
To see the output,
Here, we change the number with ord("")
Because, it is to avoid looking up the table

CHAR_RANGE = LAST_CHAR_CODE - FIRST_CHAR_CODE + 1
Add 1 because, when we do this index start with 0. So, it counts & end up being 25.

## Phase 12:
Finally, I want to interactive application & run it
I want to prompt the user to type in a message & I want to run Caesar shift on that message.
To capture user input in python, we use inbuilt function "input".

```python
message = "alpha in"
shift = 7

for x in message:
    print(x)
```

Terminal:
```
l/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
a
l
p
h
a

i
n
PS C:\Users\komal\prodigy>
```

```python
message = "alpha in"
shift = 7

for char in message:
    # convert in ASCII code
    # ORD - ordinal & it returns the unicode for char string
    char_code = ord(char)

    new_char_code = char_code + shift

    print(char, char_code, new_char_code)

```

Terminal:
```
    char_code = ord(char)
l/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
a 97 104
l 108 115
p 112 119
h 104 111
a 97 104
  32 39
i 105 112
n 110 117
PS C:\Users\komal\prodigy>
```

## Screenshot 1

```python
message = "alpha in"
shift = 7

for char in message:
    # convert in ASCII code
    # ORD - ordinal & it returns the unicode for char string
    char_code = ord(char)

    new_char_code = char_code + shift

    new_char = chr(new_char_code)

    print(char, char_code, new_char_code, new_char)
```

Terminal:
```
l/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
a 97 104 h
l 108 115 s
p 112 119 w
h 104 111 o
a 97 104 h
  32 39 '
i 105 112 p
n 110 117 u
PS C:\Users\komal\prodigy>
```

## Screenshot 2

```python
message = "alpha in"
shift = 7

#result
result = ""

for char in message:
    # convert in ASCII code
    # ORD - ordinal & it returns the unicode for char string
    char_code = ord(char)

    new_char_code = char_code + shift

    new_char = chr(new_char_code)

    result = result + new_char

    print(result)
```

Terminal:
```
l/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
h
hs
hsw
hswo
hswoh
hswoh'
hswoh'p
hswoh'pu
PS C:\Users\komal\prodigy>
```

Top window:

```python
message = "alpha in"
shift = 7

#result
result = ""

for char in message:
    # convert in ASCII code
    # ORD - ordinal & it returns the unicode for char string
    char_code = ord(char)

    new_char_code = char_code + shift

    new_char = chr(new_char_code)

    result = result + new_char

print(result)
```

Terminal (top):
```
> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
hswoh'pu
PS C:\Users\komal\prodigy> python caesar_cipher.py
hswoh'pu
PS C:\Users\komal\prodigy>
```

Bottom window:

```python
message = "alpha in"
shift = 7

#result
result = ""

for char in message:
    if char.isalpha():
        # convert in ASCII code
        # ORD - ordinal & it returns the unicode for char string
        char_code = ord(char)
        new_char_code = char_code + shift
        new_char = chr(new_char_code)
        result = result + new_char

print(result)
```

Terminal (bottom):
```
on3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
hswoh'pu
PS C:\Users\komal\prodigy> python caesar_cipher.py
hswoh'pu
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
hswohpu
PS C:\Users\komal\prodigy>
```

```python
message = "alpha in"
shift = 7

#result
result = ""

for char in message:
    if char.isalpha():
        # convert in ASCII code
        # ORD - ordinal & it returns the unicode for char string
        char_code = ord(char)
        new_char_code = char_code + shift
        new_char = chr(new_char_code)
        result = result + new_char
    else:
        result = result + char

print(result)
```

```
PS C:\Users\komal\prodigy>
 > & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
hswoh pu
PS C:\Users\komal\prodigy>
```

```python
message = "alpha in"
shift = 7

#result
result = ""

for char in message.upper():
    if char.isalpha():
        # convert in ASCII code
        # ORD - ordinal & it returns the unicode for char string
        char_code = ord(char)
        new_char_code = char_code + shift

        if new_char_code > 90:
            new_char_code -=26

        new_char = chr(new_char_code)
        result += new_char
    else:
        result += char

print(result)
```

```
u
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
hswoh pu                python caesar_cipher.py
hswoh pu
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
NY]UN V[
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
HSWOH PU
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
HSWOH PU
PS C:\Users\komal\prodigy>
```

Top editor — caesar_cipher.py:

```python
message = "alpha in!"
shift = 7

LAST_CHAR_CODE = 90
CHAR_RANGE = 26
#result
result = ""

for char in message.upper():
    if char.isalpha():
        # convert in ASCII code
        # ORD - ordinal & it returns the unicode for char string
        char_code = ord(char)
        new_char_code = char_code + shift

        if new_char_code > LAST_CHAR_CODE:
            new_char_code -= CHAR_RANGE

        new_char = chr(new_char_code)
        result += new_char
    else:
        result += char

print(result)
```

Bottom editor — caesar_cipher.py > caesar_shift:

```python
message = "alpha in!"
shift = 7

LAST_CHAR_CODE = 90
CHAR_RANGE = 26

def caesar_shift(message, shift):

    #result
    result = ""

    for char in message.upper():
        if char.isalpha():
            # convert in ASCII code
            # ORD - ordinal & it returns the unicode for char string
            char_code = ord(char)
            new_char_code = char_code + shift

            if new_char_code > LAST_CHAR_CODE:
                new_char_code -= CHAR_RANGE

            new_char = chr(new_char_code)
            result += new_char
        else:
```

Terminal output:

```
ps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
HSWOH PU
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
HSWOH PU
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
HSWOH PU
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
HSWOH PU!
PS C:\Users\komal\prodigy>
```

```python
 5    def caesar_shift(message, shift):

 7        #result
 8        result = ""

10        for char in message.upper():
11            if char.isalpha():
12                # convert in ASCII code
13                # ORD - ordinal & it returns the unicode for char string
14                char_code = ord(char)
15                new_char_code = char_code + shift

17                if new_char_code > LAST_CHAR_CODE:
18                    new_char_code -= CHAR_RANGE

20                new_char = chr(new_char_code)
21                result += new_char
22            else:
23                result += char

25        print(result)
26
27    caesar_shift("alpha in!", 7)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
HSWOH PU
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
HSWOH PU!
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
HSWOH PU!
PS C:\Users\komal\prodigy>
```

Ln 26, Col 1   Spaces: 4   UTF-8   CRLF   Python   3.11.9 64-bit (Microsoft Store)

96°F
Haze

ENG
IN

12:45
03-05-2024

**Window 2 (bottom):**

File   Edit   Selection   View   Go   Run   Terminal   Help

Welcome    caesar_cipher.py ✕

caesar_cipher.py > ...

```python
 5    def caesar_shift(message, shift):

 9
10        for char in message.upper():
11            if char.isalpha():
12                # convert in ASCII code
13                # ORD - ordinal & it returns the unicode for char string
14                char_code = ord(char)
15                new_char_code = char_code + shift

17                if new_char_code > LAST_CHAR_CODE:
18                    new_char_code -= CHAR_RANGE

20                new_char = chr(new_char_code)
21                result += new_char
22            else:
23                result += char

25        print(result)
26
27    caesar_shift("alpha in!", 7)
28    caesar_shift("copying", 7)
29    caesar_shift("engaged the co-odinates", 7)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
                    > & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
HSWOH PU!
JVWFPUN
LUNHNLK AOL JV-VKPUHALZ
PS C:\Users\komal\prodigy>
```

Ln 29, Col 38   Spaces: 4   UTF-8   CRLF   Python   3.11.9 64-bit (Microsoft Store)

96°F
Haze

ENG
IN

12:49
03-05-2024

## Top window

```python
FIRST_CHAR_CODE = 65
LAST_CHAR_CODE = 90
CHAR_RANGE = 26

def caesar_shift(message, shift):

    #result
    result = ""

    for char in message.upper():
        if char.isalpha():
            # convert in ASCII code
            # ORD - ordinal & it returns the unicode for char string
            char_code = ord(char)
            new_char_code = char_code + shift

            if new_char_code > LAST_CHAR_CODE:
                new_char_code -= CHAR_RANGE

            if new_char_code < FIRST_CHAR_CODE:
                new_char_code += CHAR_RANGE

            new_char = chr(new_char_code)
            result += new_char
```

PS C:\Users\komal\prodigy> python caesar_cipher.py
HSWOH PU!
EQRAKPI
QZSMSQP FTQ OA-APUZMFQE
PS C:\Users\komal\prodigy> & C:/Users/komal/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/komal/prodigy/caesar_cipher.py
HSWOH PU!
EQRAKPI
QZSMSQP FTQ OA-APUZMFQE
ENGAGED THE CO-ODINATES
PS C:\Users\komal\prodigy>

## Bottom window

```python
FIRST_CHAR_CODE = ord("A")
LAST_CHAR_CODE = ord("Z")
CHAR_RANGE = LAST_CHAR_CODE - FIRST_CHAR_CODE + 1

def caesar_shift(message, shift): ...

user_message = input("Message to Encrypt: ")
user_shift_key = int(input("Shift Key (integer):"))

caesar_shift(user_message, user_shift_key)
```

PS C:\Users\komal\prodigy> python caesar_cipher.py
Message to Encrypt: requesting bravo for ack
Shift Key (integer):7
YLXBLZAPUN IYHCV MVY HJR
PS C:\Users\komal\prodigy>