

Task-1

Suppose you have this email address “Amit_ml@gmail.edu”

- Input Validation: Check if the input string contains exactly one "@" symbol and at least one "." after the "@" symbol. If it's not a valid email, return "Invalid email".

```
MY_name = "Abdelhamid Ebrahim Abdelhamid"
email = "Amit_ml@gmail.edu"
[1] ✓ 0.0s

▷ 
  print(f"the number of @ in string is: {email.count('@')} and the number of . in string is: {email.count('.')}")
  if email.count("@") != 1 or "." not in email.split("@")[1]:
    |   print("Invalid email")
[4] ✓ 0.0s
...
the number of @ in string is: 1 and the number of . in string is: 1
```

Extract Username: Extract and return the part of the email before the "@" symbol.

```
|
  print(f"The username is: {email.split('@')[0]}")
] ✓ 0.0s
The username is: Amit_ml
```

Extract Domain: Extract and return the domain (the part between "@" and the last ".").

```
|
  print(f"The domain is: {email.split('@')[1].split('.')[0]}")
|
  ✓ 0.0s
The domain is: gmail
```

Check for Domain Ending: Check if the email ends with ".com". If it does, return "Commercial Domain". If it ends with ".edu", return "Educational Domain". Otherwise, return "Other Domain".

```
|
  if email.endswith(".com"):
    |   print("Commercial Domain")
  elif email.endswith(".edu"):
    |   print("Educational Domain")
  else:
    |   print("Other Domain")
]
✓ 0.0s
Educational Domain
```

Task-2:

Encoded_Message:

```
###!@mocleW EPGTQ!!!6789
```

Steps to Decode:

```
Encoded_Message= "###!@mocleW EPGTQ!!!6789"
```

1. Extract the core part of the message: "mocleW EPGTQ".

```
print(Encoded_Message[6:-7])
```

2. Reverse the first word: "mocleW" becomes "Welcom".

```
print(Encoded_Message[6:-12][::-1])
```

3. Replace shifted vowels in the second word:
 - "EPGTQ": No vowels to change. ??????

4. Final decoded message: "Welcom PGTQ".

The screenshot shows a Jupyter Notebook cell with the following code:

```
print(Encoded_Message[6:-7])
print(Encoded_Message[6:-12][::-1])
print(Encoded_Message[6:-12][::-1] + " " + Encoded_Message[-11:-7])
```

Below the code, the output is shown in a blue-bordered box:

```
5] ✓ 0.0s
```

The output consists of three lines:

- mocleW EPGTQ
- Welcom
- Welcom PGTQ

Task-3:

EncodedMessage:

```
&&&**$gnirts PLIO!!@1234
```

Steps to Decode:

1. Extract the core part of the message: "gnirts PLIO".

```
Enc_Msg= "&&&**$gnirts PLIO!!@1234"  
print(Enc_Msg[6:-7])
```

✓ 0.0s

```
gnirts PLIO
```

2. Reverse the first word: "gnirts" becomes "String".

```
print(Enc_Msg[6:-11][::-1])
```

✓ 0.0s

```
String
```

3. Replace shifted vowels in the second word:

- o "PLIO": Replace I->E and O->U to get "PLEU".

Final decoded message: "String PLEU".

```
Target = "gnirts PLIO"  
word1, word2 = Target.split()  
word1 = word1[::-1]  
word2 = word2.replace("I", "E").replace("O", "U")  
print(word1, word2)
```

✓ 0.0s

```
String PLEU
```

Task-4:

Encoded Message:

```
####@!yalpstcejorp EPUVT****9887
```

Steps to Decode:

1. Extract the core part of the message: "yalpstcejorp EPUVT".

```
E_Message= "####@!yalpstcejorp EPUVT****9887"
print(E_Message[7:-8])
```

✓ 0.0s

```
yalpstcejorp EPUVT
```

2. Reverse the first word: "yalpstcejorp" becomes "projectplay".

```
print([E_Message[7:-14][::-1]])
```

✓ 0.0s

```
projectsplay
```

3. Replace shifted vowels in the second word:

- o "EPUVT": Replace E->A, U->O to get "APOVT".

Final decoded message: "projectplay APOVT". The correct "projectsplay APOVT"

```
Target = "yalpstcejorp EPUVT"
word1, word2 = Target.split()
word1 = word1[::-1]
word2 = word2.replace("E", "A").replace("U", "O")
print(word1, word2)
```

✓ 0.0s

```
projectsplay APOVT
```