

```
Entrée [1]: import re
import os
from datetime import datetime as dt
from dateutil.parser import parse
```

Q1) Write a Python program that matches a string that has an a followed by zero or more b's

```
Entrée [ ]: ''' The function i built intends to find a non case sensitive pattern of ab or a0 i.e "a" followed by zero or more "b" '''

def str_match(text1):
    pattern1 = 'ab*?'
    ''' "*" quantifier used to find 0 or more instances, likewise, "?" used to find 0 or one instance of the pattern
    being searched '''
    if re.search(pattern1 , text1, re.IGNORECASE):
        return 'Matched'
    else:
        return('No match found')
print(str_match(input('enter charaters to match \n')))
# r used for getting the raw string from the input
```

Q2) Write a Python program to find sequences of one upper case letter followed by lower case letters

```
Entrée [ ]: def case_Match(text2):
    pat2 = '[A-Z]?[a-z]+$' #[a-z]+_[a-z]+$'
    # "?" for finding just one instance of upper case letter and "+" used to find one or more instance of lower
    # case alphabets "$" used to find match at the end
    if re.search(pat2, text2):
        return 'Matched!'
    else:
        return('No match found!')

print(case_Match(input('enter stuff \n')))
```

Q3) Write a Python program that matches a string that has an 'a' followed by anything, ending in 'b'

```
Entrée [ ]: def startA_EndB(text3):
    pat3 = '^a.*?b$'
    if re.search(pat3, text3, re.IGNORECASE):
        return 'Match found'
    else:
        return 'No Match found'
print(startA_EndB(input('Enter stuff')))
```

Q4) Write a Python program that matches a word containing 'z', not start or end of the word

```
Entrée [ ]: def containZ(text4):
    pat4 = '\Bz\B'
    # \B gives cases when Z is not at a word boundary
    if re.search(pat4, text4, re.IGNORECASE): #ignores case, remove this arg to find cases of only lower case "z"
        return 'Match Found'
    else:
        return 'No Match Found'
print(containZ(input('Enter text to check if it contains the alphabet "Z" (non case sensitive)')))
```

Q4) Write a Python program to split a string with multiple delimiters

```
Entrée [ ]: #check Done
str = 'split, string; with*multiple\ndelimiters'
re.split('; |, |\*|\n',str)
```

Write a Python program to find all adverbs and their positions in a given sentence (do this exercise for English adverbs and French adverbs)

```
Entrée [2]: sample_English = "Python is particularly used as a scripting language to automate simple but tedious tasks"
for m in re.finditer(r"\w+ly", sample_English):
    print('%d-%d: %s' % (m.start(), m.end(), m.group(0)))
# The numbers in the output correspond to the indexes of the words
```

10-22: particularly

```
Entrée [3]: sample_French = '''Python est cependant particulièrement utilisé comme langage de script pour automatiser
des tâches simples mais fastidieuses'''
for m in re.finditer(r"\w+ment", sample_French):
    print('%d-%d: %s' %(m.start(), m.end(), m.group()))
# The numbers in the output correspond to the indexes of the words
```

21-37: particulièrement

Q7) write a Python program to find all Dates in a text (Both French Format and English Format )

```
Entrée [ ]: str = "Hey Python, Find the date, it's here ---> 2018-12-10 <--- here"
match = re.search('\d{4}-\d{2}-\d{2}', str)
date = dt.strptime(match.group(), '%Y-%m-%d').date()
print(date)
```

```
Entrée [ ]: #french
parse('aujourd\'hui est la 1er Février 2019 ', fuzzy=True)
#gives date in yyyy mm dd format
```

```
Entrée [ ]: # english
parse('today is 1st February 2019 ', fuzzy=True)
#gives date in yyyy mm dd format
```

Entrée [ ]: