Machine Learning Lab - 1

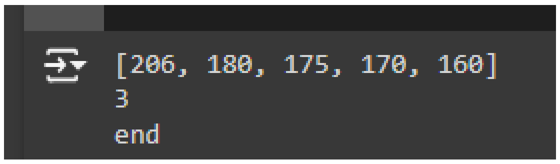
WRITE THE PYTHON CODE TO SOLVE THE FOLLOWING PROBLEMS

1. You have a group of students that were randomly assigned to dorm rooms. You wish to re-assign them to rooms such that the students are arranged in decreasing height order as you go down the hall. Each student has a lot of personal stuff to move when switching rooms, so you want to minimize the number of moves, however you don't care how far any particular move is.

Code :

```
# 1. Sorting heights

# Minimum moves should be there
heights = [160, 170, 180, 175, 206]
n = len(heights)

swaps = 0

# Selection Sort
for i in range(0, n):
  max = -1
  maxIndex = -1
  for j in range(i, n):
    if max < heights[j]:
      max = heights[j]
      maxIndex = j
  if heights[i] != max:
    temp = heights[i]
    heights[i] = heights[maxIndex]
    heights[maxIndex] = temp
    swaps += 1

print(heights)
print("Number of Swaps : ", swaps)
```

Output :



```
[206, 180, 175, 170, 160]
3
end
```

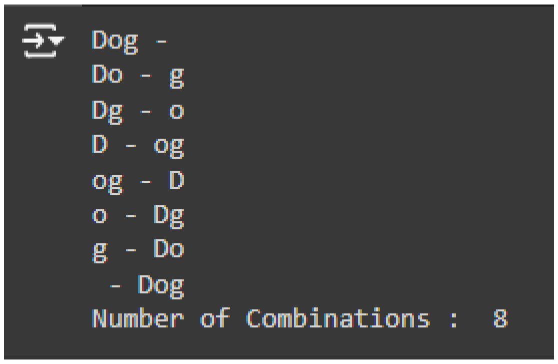
Udit Jain
24CSM1R23

2. Write a recursive function divide that uses backtracking to divide the letters from word w into two words s1 and s2. Each letter from w must be used exactly once either in s1 or s2, and the relative order of the letters must be preserved, meaning that s1 and s2 must both be subsequences of w. For example, the word “friendly”; can be divided into “find” + “rely”;

Code :

```
# 2. Split the string Recursively

def splitString(str):
  count = [0]
  splitHandler(str, 0, "", "", count)
  print("Number of Combinations : ", count[0])

def splitHandler(str, index, s1, s2, count):
  n = len(str)
  if index == n:
    print(s1, s2, sep=" - ")
    count[0] += 1
    return

  ch = str[index]
  splitHandler(str, index + 1, s1 + ch, s2, count)
  splitHandler(str, index + 1, s1, s2 + ch, count)

splitString("Dog")
```

Output :



```
Dog - 
Do - g
Dg - o
D - og
og - D
o - Dg
g - Do
 - Dog
Number of Combinations :  8
```

3. People around the globe download audio and video files on a daily basis. Web sites like movierulz, youtube, myspace, and itunes all offer a variety of digital media products. You’ve decided to put your programming skills to work and write a function to concurrently

Udit Jain
24CSM1R23

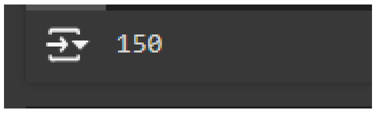
download your full wish list of songs and movies. The following routine is thread-safe and has already been written for you. It downloads the contents of the specified file from the specified server and returns the total number of bytes downloaded. Your job is to implement the DownloadMediaLibrary function, which takes a server name and an array of file names, downloads all files using the DownloadMediaFile routine, and returns the total number of bytes downloaded. For each file in the array, you should spawn off a separate thread to download its contents. However, because you don't want to pelt the file server with an unreasonable number of simultaneous requests, you should limit the number of active connections to 3, so that the number of concurrently executing calls to DownloadMediaFile will never be more than 2 at any single instant. You must ensure that all calls to DownloadMediaFile have completed before DownloadMediaLibrary returns.

Code :

```
# 3. Size Sum

import time

requests = 0

def api(serverName, fileName):
  requests += 1
  files = {'a' : 10, 'b' : 20, 'c' : 30, 'd' : 40, 'e' : 50}
  # api fetching work
  # ...
  if requests > 2:
    # while api fetching work done for any of the previous two threads :
    time.sleep(1)
  return files[fileName]

def downloadMediaLibrary(serverName, filesName):
  sum = 0
  for file in filesName:
    sum += api(serverName, file)
    time.sleep(1)
  print(sum)

downloadMediaLibrary("https://...", ["a", "b", "c", "d", "e"])
```

Output :

Udit Jain
24CSM1R23



```
150
```

4. In the early part of the 20th century, there was considerable interest in both England and the United States in simplifying the rules used for spelling English words, which has always been a difficult proposition. One suggestion advanced as part of this movement was the removal of all doubled letters from words. If this were done, no one would have to remember that the name of the Stanford student union is spelled “Tresidder” even though the incorrect spelling “Tressider” occurs at least as often. If double letters were banned, everyone could agree on “Tresider”. Write a method removeDoubledLetters that takes a string as its argument and returns a new string with all doubled letters in the string replaced by a single letter. For example, if you call

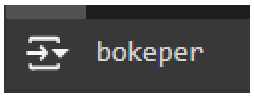
removeDoubledLetters(&quot;bookkeeper&quot;)

Your method should return &quot;bokeper&quot;.

Code :

```
# 4. Remove repetations

def removeDoubledLetters(str):
  n = len(str)

  if n == 0:
    print("Empty String")
    return

  res = ""
  ch = str[0]
  i = 1
  while i < n:
    if ch != str[i]:
      res += ch
      ch = str[i]
    i += 1
  res += ch
  print(res)

removeDoubledLetters("bookkeeper")
```

Output :



```
bokeper
```

Udit Jain
24CSM1R23

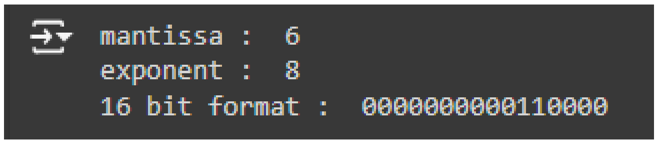
5. 8-bit words are used to represent a certain set of natural numbers including zero. The least-significant 3 bits contain an unsigned binary-encoded mantissa value. The remaining 5 bits represent an unsigned binary-encoded, bitwise left shift to be applied to the mantissa to obtain the represented value. Give two functions that respectively convert an encoded value to its

a. nearest 16-bit unsigned integer

Code :

```
# 5. encoding 8 to 16 bits and double precision

# A. Conversion into 16 bit

num = 30

mantissa = num & 7
exponent = 2 ** ((num & 120) >> 3)

print("mantissa : ", mantissa)
print("exponent : ", exponent)

decimal = mantissa * exponent

hex = ""
num = int(decimal)
while num != 0:
  rem = num & 1
  hex = str(rem) + hex
  num >>= 1

hex = ((16 - len(hex)) * '0') + hex
print("16 bit format : ", hex)
```

Output :



```
mantissa :  6
exponent :  8
16 bit format :  0000000000110000
```

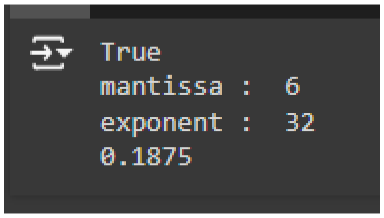
b. nearest double-precision floating-point number.

Udit Jain
24CSM1R23

Code :

```
# 5.B Conversion into double precision

num = 30 + 16 + 128

isExNeg = ((num & 128) != 0)
if isExNeg:
  num -= 128

mantissa = num & 7
exponent = 2 ** ((num & 120) >> 3)

print("mantissa : ", mantissa)
print("exponent : ", exponent)

floatNum = float(mantissa) * ((1.0/float(exponent)) if isExNeg else float(exponent))

print(floatNum)
```

Output :



```
True
mantissa :  6
exponent :  32
0.1875
```

Udit Jain
24CSM1R23