UNIVERSITY OF SOUTH DAKOTA

Assignment 01

CSC557- Data Analysis Decision Making and Data Visualization

September 16, 2021

Md Minhazur Rahman

ID: 101085958

MS in Physics-Analytics for Large Data Sets

Department of Physics,

University of South Dakota.

Problem no: 1

a.

In [47]:
```
import sys
from IPython import get_ipython


## This function will be used to validate input number

def getNumber():
    pid = os.getpid()
    try:
        return int(input().strip())
    except ValueError:
        print('Please enter a valid number')
    sys.exit('sfd')


print('Please enter the size of your container')
size = getNumber()

if size<=1:
    print('You will receive {:.2f}$'.format(0.10))
else:
    print('You will receive {:.2f}$'.format(0.25))
```

```
Please enter the size of your container
4
You will receive 0.25$
```

b.

In [48]:
```
print('Please enter the number')

num = getNumber()

if num%2==0:
    print('The number is even')
else:
    print('The number is odd')
```

```
Please enter the number
3
The number is odd
```

Problem no: 2

a.

In [49]:
```
## this function will determine how to give minimum number of coins
## I have done all kind of validation
def chagneCalculator(change):
    message = ''
    if int(change/2) or change==2 :
        toonie = int(change/2)
        change = change - toonie*2
```

```
        message += '{} Toonies '.format(toonie)

    if change >= 1:
        change = change - 1
        message += '{} Loonies '.format(1)

    if (change%0.25) or change==0.25:
        quarter = int(change/0.25)
        change = round(change - quarter*0.25,2)
        message += '{} Quarter '.format(quarter)

    if int(change/0.1) or change==0.1:
        dime = int(change/0.1)
        change = round(change - dime*0.1,2)
        message += '{} Dime '.format(dime)

    if int(change/0.05) or change==0.05:
        nickel = int(change/0.05)
        change = round(change - nickel*0.05,2)
        message += '{} Nickel '.format(nickel)

    if change>0:
        penny = int(change*100)
        message += '{} Penny '.format(penny)

    return message

try:
    print('Please enter your total cost')
    cost = float(input().strip())
    print('Please enter your cash amount to pay')
    payment = float(input().strip())

    change = payment-cost

    ## chekc if user inserts any negative number
    if cost<0 or payment<0:
        print('Enter positive number')

    ## chekc if cost is greater than payment
    elif change<0:
        print('Cost can not be greater than payment')
    else:
        print('You will receive:\n')
        print(chagneCalculator(change))

except ValueError:
    print('Please enter a valid number')
```

```
Please enter your total cost
5.33
Please enter your cash amount to pay
10
You will receive:

2 Toonies 2 Quarter 1 Dime 1 Nickel 2 Penny
```

b.

In [50]:

```
print('Enter 1st number')
num1 = int(getNumber())
print('Enter 2nd number')
num2 = int(getNumber())
```

```
print('Enter 3rd number')
num3 = int(getNumber())

nums = [num1,num2,num3]
nums.sort()
print('Sorted list:')
print(nums)

print('Minimum number of the list:')
print(min(nums))
print('Maximum number of the list:')
print(max(nums))
```

```
Enter 1st number
4
Enter 2nd number
3
Enter 3rd number
8
Sorted list:
[3, 4, 8]
Minimum number of the list:
3
Maximum number of the list:
8
```

Problem no: 3

In [51]:

```
print('Enter the data. ex: September 17')
from datetime import datetime

x = str(input()).strip()

## Function to convert string to date

def dateConvert(x):
    ## For input validation i used try except block. If someone enters date i
    ## it will notify the user.

    try:
        return datetime.strptime(x, '%B %d')
    except ValueError:
        print('Enter your data according to given format. ex: September 17')

date1 = dateConvert('January 1')
date2 = dateConvert('March 20')
date3 = dateConvert('June 21')
date4 = dateConvert('September 22')
date5 = dateConvert('December 21')

## Function for finding the season

def findSeason(date):
    if date>=date1 and date<=date2:
        print('Spring')
    elif date>=date2 and date<=date3:
        print('Summer')
    elif date>=date3 and date<=date4:
        print('Fall')
    elif date>=date4 and date<=date5:
        print('Winter')
```

```
date = dateConvert(x)
findSeason(date)
```

```
Enter the data. ex: September 17
July 3
Fall
```

Problem no: 4

In [53]:

```
print('Rating should be 0.0, 0.4, 0.6 or more')
print('Please enter your rating')

def ratingLabeler(rating):
    if rating==0.0:
        return 'Unacceptable'
    elif rating==0.4:
        return 'Acceptable'
    else:
        return 'Maritorius'


try:
    rating = float(input())
    if rating == 0.0 or rating == 0.4 or rating >=0.6:
        print('The employee\'s performance is '+ ratingLabeler(rating))
        print('His salary should be raised by {:.2f}$'.format(rating*2400))
    else:
        print('Enter a valid rating')
except ValueError:
    print('Please enter a valid rating')
```

```
Rating should be 0.0, 0.4, 0.6 or more
Please enter your rating
1
The employee's performance is Maritorius
His salary should be raised by 2400.00$
```

Problem no: 5

In [16]:

```
import re,sys

print('Please enter your password')
password = input()
lower = '[a-z]+'
upper = '[A-Z]+'
number = '[0-9]+'
if len(password)<8:
    print('Password must contain 8 characters')
    #sys.exit(0)
if len(re.findall(lower,password))==0:
    print('Password must contains at least one lower case characters')
if len(re.findall(upper,password))==0:
    print('Password must contains at least one upper case characters')
if len(re.findall(upper,password))==0:
    print('Password must contains at least one number')
```

```
Please enter your password
hj
Password must contain 8 characters
```

```
Password must contains at least one upper case characters
Password must contains at least one number
```

In [ ]: