Triangle decision problem

```
a = int(input())
b = int(input())
c = int(input())

if a < b+c and b < a+c and c < a+b:
    istr = 'y'
else:
    istr = 'n'

if(istr == 'y'):
    if(a== b) and(b == c):
        print("Equilateral triangle")
    elif(a!=b) and (a!=c) and (b!=c):
        print("scalene triangle")
    else:
        print("isosceles triangle")
else:
    print("Not a triangle")
```

Triangle boundry and equi problem

```
a = int(input())
b = int(input())
c = int(input())

c1 = a >= 1 and a <= 10
c2 = b >= 1 and b <= 10
c3 = c >= 1 and c <= 10
res = "n"
while(not (c1 and c2 and c3)):
    if(not c1):
      print("The value of a is not permitted")

    if(not c2):
      print("The value of b is not permitted")

    if not c3:
      print("The value of c is not permitted")
    res = "y"
    break

if(res == "n"):
    if a < b+c and b < a+c and c < a+b:
        istr = 'y'
    else:
        istr = 'n'
```

```
    if(istr == 'y'):
        if(a== b) and(b == c):
            print("Equilateral triangle")
        elif(a!=b) and (a!=c) and (b!=c):
            print("saclene triangle")
        else:
            print("isosceles triangle")
    else:
        print("Not a triangle")
```

commission

```
lock = int(input())
stock = int(input())
barrels = int(input())

total_sum = 0
if (0 < lock <= 70) and (0 < stock <= 80) and (0 < barrels <= 90):
    a = int(input())
    b = int(input())
    c = int(input())

    if (0 < a <= lock) and (0 < b <= stock) and (0 < c <= barrels):
        total_sum = (45 * a) + (30 * b) + (25 * c)

        if total_sum <= 1000:
            res =total_sum * 10 // 100
        elif 1001 <= total_sum <= 1800:
            res =total_sum * 15 // 100
        else:
            res =total_sum * 20 // 100

        print("Total Sale is Rs.", total_sum, "\nThe commission is Rs.",
res)
    else:
        print("Sale Exceeds")
else:
    print("Limit Exceeds")
```

binary search

```
def binarySearch(array, x, low, high):
    while low <= high:
        mid = (low + high) //2
        if array[mid] == x:
            return mid
        elif x > array[mid]:
            low = mid + 1
```

```
        else:
            high = mid - 1
    return -1
array = input().split(" ")
x = input()
result = binarySearch(array, x, 0, len(array)-1)
if result != -1:
    print("Element is present at index ",str(result))
else:
    print("Not found")
```

insertion sort

```
def insertionSort(arr, n):
    for i in range(1,n):
        k = arr[i]
        j = i - 1
        while j >= 0 and k < arr[j]:
            arr[j+1] = arr[j]
            j -= 1
        arr[j+1] = k

lst = []
n = int(input())
for i in range(n):
    lst.append(int(input()))
insertionSort(lst, n)
print(lst)
```

quick sort

```
def pivot_fun(lst, i, l):
    pivot = lst[i]
    left = i+1
    right = l
    while True:
        while (left <= right) and (lst[left] <= pivot):
            left += 1
        while (left <= right) and (lst[right] >= pivot):
            right -= 1
        if right < left:
            break
        else:
            lst[left],lst[right] = lst[right], lst[left]
    lst[i], lst[right] = lst[right], lst[i]
    return right

def quicksort(lst, i , j):
    if i < j:
```

```
        p = pivot_fun(lst, i, j)
        quicksort(lst, i, p-1)
        quicksort(lst, p+1, j)

lst = []
l = int(input())
for i in range(l):
    lst.append(int(input()))
quicksort(lst,0, l-1)
print(lst)
```

next date boundry , equi

```
def leap(year):
    if year % 4 == 0 and year % 100 != 0 or year % 400 == 0:
        return True
    else:
        return False


def check(day, month):
    if (month == 4 or month == 6 or month == 9 or month == 11) and day ==
31:
        return True
    else:
        return False


print()
day = int(input())
month = int(input())
year = int(input())
flag = 'y'
tom_day = day
tom_month = month
tom_year = year
if day < 1 or day > 31:
    print("Invalid Day")
    flag = 'n'
elif month < 1 or month > 12:
    print("Invalid Month")
```

```
    flag = 'n'
if check(day, month):
    print("Invalid Day and Month")
    flag = 'n'
elif year < 1812 or year > 2023:
    print("Invalid Year")
    flag = 'n'
if month == 2:
    if leap(year) and day > 29:
        print("Invalid input for Leap Year")
        flag = 'n'
    elif not leap(year) and day > 28:
        print("Invalid input for non Leap Year")
        flag = 'n'
if flag == 'y':
    if month == 1 or month == 3 or month == 5 or month == 7 or month == 8
or month == 10:
        if day < 31:
            tom_day = day + 1
        else:
            tom_day = 1
            tom_month = month + 1
    if month == 4 or month == 6 or month == 9 or month == 11:
        if day < 30:
            tom_day = day + 1
        else:
            tom_day = 1
            tom_month = month + 1
    if month == 12:
        if day < 31:
            tom_day = day + 1
        else:
            tom_day = 1
            tom_month = 1
            tom_year = year + 1
```

```
    if month == 2:
        if day < 28:
            tom_day = day + 1
        elif leap(year) and day == 28:
            tom_day = day + 1
        elif day == 28 or day == 29:
            tom_day = 1
            tom_month = 3
a= []
a.append(str(tom_day))
a.append(str(tom_month))
a.append(str(tom_year))
if flag == 'n':
    print("Enter all the valid entries")
else:
    print("Next Date is: ", "/".join(a))
```

grading

```
arr = []
for i in range(5):
   arr.append(int(input()))
sum = sum(arr)
per = sum/5
print(per)
if per>90:
    print("Destinction")
elif per > 75:
    print("First class")
elif(per >50):
    print("Good")
else:
    print("Fail")
```