In [1]:

```
#quest1

list=[]
x=int(input("print the number of the elments of the list"))
for i in range (x):
 z=int(input("print the element of the list"))
 list.append(z)

def binary_search(A, l, h, k):
    if h >= l:
        mid = int(l + (h - l)/2)

        if A[mid] == k:
            return "true"
        elif A[mid] > k:
            return binary_search(A, l, mid-1, k)
        else:
            return binary_search(A, mid+1, h, k)
    else:
        return 'false'
k=5
l=0
h=len(list)-1
binary_search(list,l,h,k)
```

```
print the number of the elments of the list3
print the element of the list4
print the element of the list6
print the element of the list5
```

Out[1]:

```
'false'
```

In [2]:

```
#quest2
def power(a,b):
 if b==0:
  return 1
 elif a==0:
   return 0
 elif b==1:
  return a
 else:
  return a*power(a,b-1)

print(power(3,4))
```

```
81
```

In [ ]:

```
#quest3
list=[]
x=int(input("print the number of the elments of the list"))
for i in range (x):
 z=int(input("print the element of the list"))
 list.append(z)
def bubbleSort(nlist):
 for passnum in range(len(nlist)-1,0,-1):
  for i in range(passnum):
   if nlist[i]>nlist[i+1]:
    temp = nlist[i]
    nlist[i] = nlist[i+1]
    nlist[i+1] = temp
 return nlist
```

In [ ]:

```
#quest4
list=[]
x=int(input("print the number of the elments of the list"))
for i in range (x):
 z=int(input("print the element of the list"))
 list.append(z)
    def mergeSort(myList):
    if len(myList) > 1:
        mid = len(myList) // 2
        left = myList[:mid]
        right = myList[mid:]


        mergeSort(left)
        mergeSort(right)


        i = 0
        j = 0


        k = 0

        while i < len(left) and j < len(right):
            if left[i] < right[j]:

              myList[k] = left[i]

              i += 1
            else:
                myList[k] = right[j]
                j += 1

            k += 1


        while i < len(left):
            myList[k] = left[i]
            i += 1
            k += 1

        while j < len(right):
            myList[k]=right[j]
            j += 1
            k += 1
```

In [ ]:

```
#quest5
list=[]
x=int(input("print the number of the elments of the list"))
for i in range (x):
 z=int(input("print the element of the list"))
 list.append(z)
def partition(array, start, end):
 pivot = array[start]
 low = start + 1
  high = end
  while True:
   while low <= high and array[high] >= pivot:
    high = high - 1
    while low <= high and array[low] <= pivot:
     low = low + 1
      if low <= high:
       array[low], array[high] = array[high], array[low]
        else:
            break
array[start], array[high] = array[high], array[start]

return high
```