CS263 LAB3

Name: Dipean Dasgupta **Section:2B**
Roll:202151188

Task: Create a program to find maximum number of people killed with strength.

Solution Code:

```
import java.util.*;
public class maxkill {
    public static int Killcount(int S,int N){            //function for counting people killed
        int []sum = new int[N];                          //array declared for storing sum
        sum[0] = 0;                                      //sum initialized to 0 as sq(0)=0
        for (int i = 1; i < N; i++){
            sum[i] = (i * i) + sum[i - 1];               //array of calculated sum created
        }
        int X = lb(sum, l: 0, N, S);                     //calling lb function to find element X which
        if (sum[X] > S){                                 //is greater than or equal to given value
            --X;                                         //if sum of X elemnts> strength; decrement by 1
        }
        return X;                                        // return X as number of people killed
    }
    public static int lb(int[] A, int l,int h, int e){
        if(l==h){
            return -1;
        }else{
        while(l < h){
            int md = l + (h - l)/2;                      // lb function here is A binary search function
            if(e > A[md])                                //to find the element greater than or
                l = md + 1;                              //equal to given value
            else
                h = md;
            }
        }
        return l;
    }
```

```
    Run | Debug
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        System.out.println(x: "Enter Strength Value of killer:");
        int S=sc.nextInt();                                     //scanning input of strength value
        System.out.println(x: "Enter number of people:");
        int N=sc.nextInt();                                     //scanning input of people number
        System.out.println(x: "Max number of people killed:");
        System.out.println(Killcount(S,N));                     // applying killcount function
        sc.close();
    }
}
```

OUTPUT:

```
PS D:\Java\ALCS263> cd "d:\Java\ALCS263\" ; if ($?) { javac maxkill.java } ; if ($?) { java maxkill }
Enter Strength Value of killer:
14
Enter number of people:
10
Max number of people killed:
3
```

```
PS D:\Java\ALCS263> cd "d:\Java\ALCS263\" ; if ($?) { javac maxkill.java } ; if ($?) { java maxkill }
Enter Strength Value of killer:
1650
Enter number of people:
1000
Max number of people killed:
16
```

Time complexity:

Here as we can see if we kill I indexed person then we already killed (i-1) indexed person. So, firstly the element greater than equal to value is found using binary search which completes the task in O(log(N)). Then the cumulative sum is calculated up to the element returned by lb function. Comparing it with Strength S, if S< sum then reduced element value by 1 and return the value which will be the desired result.

So

Time complexity: O(log(N))