# LAB-4

Name: V.SAIKRISHNA

**Reg.No.:** 19BCE7638

### 1. Maximum Salary

```
import java.util.*;

public class Main {
    private static String largestNumber(String[] salaryParts) {
        int numParts = salaryParts.length;
        if (salaryParts == null || numParts == 0)
            return "";

        String[] maxSalary = new String[numParts];
        for (int i = 0; i < numParts; ++i) {
            maxSalary[i] = String.valueOf(salaryParts[i]);
        }

        Arrays.sort(maxSalary, (s1, s2) -> (s2 + s1).compareTo(s1 + s2));
}
```

```
StringBuilder sb = new StringBuilder();
  for (String salaryPart : maxSalary) {
    sb.append(salaryPart);
  }
  return sb.toString();
}
public static void main(String[] args) {
  Scanner scanner = new Scanner(System.in);
  int n = scanner.nextInt();
  String[] salaryParts = new String[n];
  for (int i = 0; i < n; i++) {
    salaryParts[i] = scanner.next();
  }
  System.out.println(largestNumber(salaryParts));
}
```

### **Output:**

}

```
2
21 2
221

...Program finished with exit code 0
Press ENTER to exit console.
```

```
5
9 4 6 1 9
99641
...Program finished with exit code 0
Press ENTER to exit console.
```

```
3
23 39 92
923923
...Program finished with exit code O
Press ENTER to exit console.
```

## 2.Car fuelling problem:

```
import java.util.*;
import java.lang.*;
```

```
import java.io.*;
class Main
{
  static int compute refills(int dist,int tank,int stops[],int n){
     int current refills=0;
     int num refills=0;
     int last refill=0;
     while(current_refills<=n) {</pre>
       last refill = current refills;
       while ((current refills <= n) && (stops[current refills + 1] -
stops[last_refill]) <= tank) {</pre>
          current refills = current refills + 1;
       }
       if (current_refills == last_refill)
          return -1;
       if (current refills <= n)</pre>
          num refills = num refills + 1;
     }
     return num refills;
  }
```

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int dist = scanner.nextInt();
    int tank = scanner.nextInt();
    int n = scanner.nextInt();
    int stops[] = new int[n+2];
    stops[0] = 0;
    stops[n+1] = dist;
    for (int i = 1; i \le n; i++) {
       stops[i] = scanner.nextInt();
    }
    System.out.println(compute_refills(dist,tank,stops,n));
  }
}
```

### **Output:**

```
input

950

400

4

200 375 550 750

2

...Program finished with exit code 0

Press ENTER to exit console.
```

```
10
3
4
1 2 5 9
-1
...Program finished with exit code 0
Press ENTER to exit console.
```

#### **Analysis:**

#### 1. Maximum Salary problem:

```
Public claro Main &
  private static String Largest Neumber (String 1) salay Parts) &
       int num Parts = salary Parts length; -> 1
    if (salay parts = = null 11 nuthernotes) } = 0
        string [] mansalary = new string [numparts];
for lint i = 0; i z num Parts; ++i) f
             man salary (i) = String, value of (salary facts (1)),
  (a) a 3. Fe har a William tool James
     Arrays sort (marsalay, (5,,52) -> (52+5).compacto (5,+52))).
       StringBullder & b = new stringBuilder();
                                                  nlogn
       for (string salay Part: max salay) ;
                                                     O(nlogn)
              Sb. append (salary part), 30 time
           setur sb. tosting (),
```

2.	Car	fuel	ling	prob	olem:
----	-----	------	------	------	-------

int num refills = 0; intilast nefill =0; dragmun in while (unent\_refills = n) of - s(n+1) last refills = anest refills, -> n while ( comment - refills 2 = n ) 68 (stops coment - ref massalong Cit = String, ratur of (salong fachetis), Stop[last-refill] == fant ) { -> n(n) ament refills = ament refills +1, schaller she am stirelling if (ament-refiles = cost refill) of retire - 1 jobs & brigger de