PRACTICAL: 06

AIM: Implementation of solution of Activity Selection Problem using Greedy method.

ALGORITHM: ___________________________

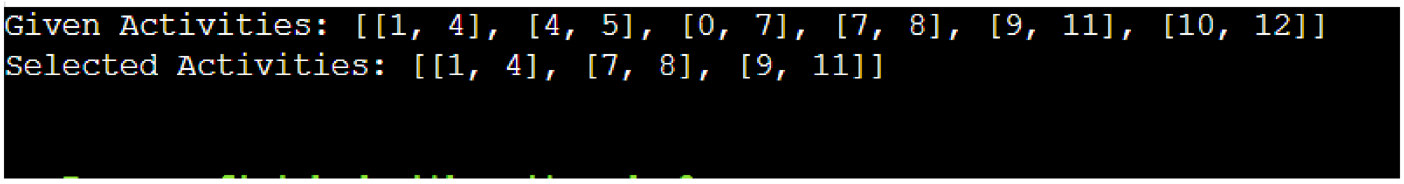
```
1. Sort all the activities according to their end time.
2. Select the first activity and note the end time, call it as current_end.
3. Now iterate through the rest of the activities. For each current_activity
        1. If start time of current_activity > current_end
                1. Select current_activity.
                2. Update current_end = end time of current_activity.
    2. Else
        1. Ignore the current_acitvity.
```

CODE: ___________________________

```
#include <bits/stdc++.h>

using namespace std;



void printMaxActivities(int s[], int f[], int n)

{

   int i, j;

   cout <<"Following activities are selected "<< endl;

   i = 0;

   cout <<" "<< i;

   for (j = 1; j < n; j++)

   {

     if (s[j] >= f[i])

     {

         cout <<" " << j;

         i = j;

     }

   }

}

{

   int s[] =  {1, 3, 0, 5, 8, 5};

   int f[] =  {2, 4, 6, 7, 9, 9};

   int n = sizeof(s)/sizeof(s[0]);
```

```
    printMaxActivities(s, f, n);
    return 0;
}
```

OUTPUT: _____________________________



```
Given Activities: [[1, 4], [4, 5], [0, 7], [7, 8], [9, 11], [10, 12]]
Selected Activities: [[1, 4], [7, 8], [9, 11]]
```

TIME COMPLEXITY:-

<u>Best Case</u>:-

O(n)

<u>Average Case</u>:-

O(n*log n)

<u>Worst Case</u>:-

O(n*log n)