

# Design and Analysis of Algorithms Assignment - 7

Name: Dhanraj Kore

Div: TY B

Roll No: 60

Batch : B-3

## Conflict free Job Scheduling

### CODE :

```
#include<bits/stdc++.h>
using namespace std;

bool helper(pair<int,int> p1,pair<int,int> p2)
{
    return p1.second<p2.second;
}

int main()
{
    int n;
    cout<<"Enter the no. of jobs : ";
    cin>>n;
    vector<pair<int,int>> v(n);
    cout<<"Enter the starting time and ending time of each job: "<<endl;
    for(int i=0;i<n;i++)
    {
        cout<<"Job "<<i<<" : ";
        cin>>v[i].first>>v[i].second;
    }
    sort(v.begin(),v.end(),helper);
    cout<<"The job sequence should be : "<<endl;
```

```

cout<<"Starting Time "<<" Ending Time"<<endl;

cout<<v[0].first<<'\\t'<<v[0].second<<endl;

int temp = v[0].second;

for(int i=1;i<n;i++)
{
    if(v[i].first>=temp)
    {
        cout<<v[i].first<<'\\t'<<v[i].second<<endl;

        temp = v[i].second;
    }
}
}

```

O/P:

The screenshot shows the Visual Studio Code editor with a C++ file named `main.cpp` open. The code implements a selection sort algorithm to find the minimum element in an array of jobs, where each job is represented by a pair of (start time, end time). The code prints the starting and ending times of the jobs in sorted order.

The terminal output shows the execution of the program. It prompts the user to enter the number of jobs (5) and then the starting and ending times for each job. The output shows the jobs in sorted order: Job 0: 2 4, Job 1: 1 5, Job 2: 3 6, Job 3: 5 9, Job 4: 6 8. The final output shows the job sequence should be: 2 4, 1 5, 3 6, 5 9, 6 8.

Time Complexity:  $O(n \log n)$

Space Complexity:  $O(1)$