Custom comparator

↳ STL → sort()

sort(v.begin(), v.end())

↳ by default → accending order

to decending → using custom comparator

sort(v.begin(), v.end(), mycomp)

↳ custom comparator

```
bool mycomp(int &a, int &b)
{
    return a > b;
}
```

↳ decreasing order sorting

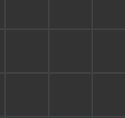
Vector [vector]

Sorting in respect of second element

```
[
  [1,44],
  [0,55],
  [0,22],
  [0,11],
  [2,33]
]
```

O/P

```
[
  [0,11],
  [0,22],
  [2,33],
  [1,44],
  [0,55]
]
```



0 → [0,11] [1,44]
1 → [0,55]
2 → [0,22]
3 → [0,11]
4 → [2,33]

Sorting according to 1st index

Sort(v.begin(), v.end())

↳ By default → Sorted with 0 index

↳ We use custom comparator

```
sort( v.begin(), v.end(), myComp )

bool myComp ( vector<int> &a, vector<int> &b )
{
    return a[1] < b[1];
}
```

[0,11] .... [0,22] ....