→ COMPARATOR

```
BOOL CMP(PAIR<INT, INT> A, PAIR<INT, INT> B)
{
    => Jis order mai chaiye use return karde
    IF (A.FIRST != B.FIRST)
    {
        RETURN A.FIRST < B.FIRST;
    }
    RETURN A.SECOND > B.SECOND;
}
```

→ comparator for an array to return in decreasing order

```
BOOL CMP_2 (INT A, INT B)
{
    RETURN A > B;
}

INT MAIN()
{
    INT N;
    CIN >> N;
    VECTOR<PAIR<INT, INT>> V_P(N);

    FOR (INT i=0; i < n; i++)
    {
        CIN >> V_P[i].FIRST >> V_P[i].SECOND;
    }
```

→ Let us say we want to sort this pair, with 1st no. in ↑sing order and if 1st no of two pairs are same then ↓sing order according to 2ND no.
For this type of operation we have to define a comparator function as a 3RD argument in SORT() function.

```
    SORT(V_P.BEGIN(), V_P.END(), CMP);

    FOR (AUTO &x : V_P)
    {
        COUT << x.FIRST << " " << x.SECOND
        << "\n";
    }
    RETURN 0;
}
```

-> Input

6
4 3
5 5
5 3
2 5 6
7 9
8 5

-> Output

4 3
5 5
5 3
7 9
8 5
2 5 6