Iterators

↓ STL ke containers ke elements ko point karega.

Vector

| 2 | 3 | 5 | 6 | 7 | 8 |

it .begin() ↓ V.begin() ↓ V[0]

.end() ↓ next to last

* declaration

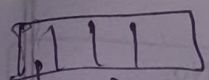
vector<int> :: iterator it = V.begin();

↓ is container ke liye iterator define kar paaeh...

→ to find value (same as value)

cout << (*it) << endl; → print 2

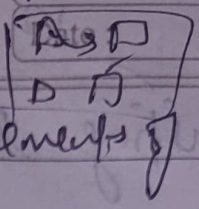
cout << (*(it+1)) << endl; → print 3



++it vs it+1 → same in case of vector, diff in map & set

move to next iteration

move to next location



* we can use iterator to iterate elements of container.

```
for(it = V.begin(); it != V.end(); ++it)
{
    cout << (*it) << endl;
}
```

* in case of pair

```
vector < pair<int,int> > V_p = {{1,2},{2,3},{3,4}};
```

ye it vector pair type hoga

```
vector< pair<int,int> > :: iterator it;
for(it = V_p.begin(); it != V_p.end(); ++it)
{
    cout << (*it) << endl;   → ye ek pair h
}
```

pair
1 2
2 3
3 4

```
cout << (*it).first << " " << (*it).second << endl;
```

(*it).first ↳ OR it -> first

it -> second

Note:-

advance (InputIterator it , distance n);

it : Iterator to be advanced

n : Number of element position to advance

→ advance advances the iterator 'it' by n element position.

eg advance(it, 2);

in cpp 11

* writing short code for iterators.

(1) Range based loops

→ instead of writing

```
for(it=v.begin(); it!=v.end(); ++it)
{
    cout<<(*it)<<endl;
}
```

→ we could write

```
for(int value:v){
    cout<<value<<" ";
}
```

ye value copy hue h reference nahi. So value mein change karenge toh V aise hi change hoga but if

```
for(int &value:v)
```

toh abb value mein change V vector ko change karega.

→ for pair

```
vector<pair<int,int>> v-p={{1,2},{2,3},{3,4},{5,6}};
for(pair<int,int> value: v-p){
cout<<value.first<<" "<<value.second<<endl;
}.
```

Page No.
Date :

② Auto Keyword → automatically data type assume kar leta h

```
vector<int>::iterator it;

for(int it = v.begin(); it != v.end(); ++it)
```

```
for(auto it = v.begin(); it != v.end(); ++it)
```

ye apne aap dekh lega ye vector of int ka iterator h

⟹ by mixing both ① & ②

initially :-

```
vector<pair<int,int>> v-p = {{1,2}, {2,3}, {3,4}};

vector<pair<int,int>>::iterator it;
for(it = v-p.begin(); it != v-p.end(); ++it)
{
    cout<<(*it).first<<" "<<(*it).second<<endl;
}
```

finally :-

```
Vector<pair<int,int>> v-p = {{1,2}, {2,3}, {3,4}};
for(auto it : v-p)
cout<<(*it).first<<" "<<(*it).second<<endl;
```