28 March 2022 11:50

C++ STL

- Standard template libraries

  i) Containers
  ii) Iterators
  iii) Algorithms
  iv) Functors

Containers- Data structures already implemented.

A) Sequential
   a. Vectors
   b. Stack
   c. Queue
B) Ordered
   a. Maps
   b. Multimaps
   c. Set
   d. Multisets
C) Unordered
   a. Unordered map
   b. Unordered set

Also team "pair"

28 March 2022 11:57

Nested containers (container inside a container)

A) Vector<vector<int>>
B) Map<int, vector<int>>
C) Set<pair<int, string>>
D) Vector<map<int, set<int>>>

28 March 2022 11:59

Iterators:

- Point to memory address of containers
- Begin(), end();
- Vector<int>:: iterator it; (how to write it short)

Continuity and discontinuity for containers

28 March 2022 12:00

Algorithms:

- Upper bound
- Lower bound
- Sort(comparator) : comparator for custom sorting behaviour
- Max element
- Min element
- Accumulate (sum of array)
- Reverse
- Count
- Find
- next permutations
- Prev permutations

28 March 2022 12:03

Functors

Classes which can act as functions

You need the knowledge of classes to understand functors.

28 March 2022 12:05

Lecture: 2

Pair

Class in c++ stl which stores two values

Let's say I want to make the pair of two data types or containers, e.g. int and string

Initializing pairs:

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



int main()
{
    pair<int, string> p;
    p = make_pair(2, "abc"); //Using inbuilt function to
                             //add element in the pair
    cout<< p.first << " " << p.second<< endl;
}
```

p.first gives us the first value
p.second gives us the second value

28 March 2022 12:13

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



int main()
{
    pair<int, string> p;
    // p = make_pair(2, "abc"); //Using inbuilt function to
                                //add element in the pair

    p = { 2, "ankit"};
    cout<< p.first << " " << p.second<< endl;
}
```

These are two ways to initialize the pair.

28 March 2022 12:14

Copying pairs.

Copying the value

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



int main()
{
    pair<int, string> p;
    // p = make_pair(2, "abc"); //Using inbuilt function to
                                 //add element in the pair

    p = { 2, "ankit"};
    // Now we can also copy the pairs just like variable

    pair<int, string> p1;
    p1=p;
    cout<< p1.first << " " << p1.second<< endl;
}
```

C:\Windows\system32\cmd.exe

```
2 ankit

Press any key to continue . . .
```

28 March 2022 12:18

Changing the value of p1 to see whether the change is reflected in p or not.

After experimentation we observe that the change is not reflected.

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



int main()
{
    pair<int, string> p;
    // p = make_pair(2, "abc"); //Using inbuilt function to
                                 //add element in the pair

    p = { 2, "ankit"};
    // Now we can also copy the pairs just like variable

    pair<int, string> p1;
    p1=p;
    p1.first= 7;
    cout<< "Value of P " << p.first << " " << p.second<< endl;
    cout<< "Value of P1 " << p1.first << " " << p1.second<< endl;
}
```

C:\Windows\system32\cmd.exe

```
Value of P 2 ankit
Value of P1 7 ankit

Press any key to continue . . .
```

28 March 2022 12:19

So how can we make change in original value of p by making changes in p1,

The way out is "Reference"

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



int main()
{
    pair<int, string> p;
    // p = make_pair(2, "abc"); //Using inbuilt function to
                                //add element in the pair

    p = { 2, "ankit"};
    // Now we can also copy the pairs just like variable

    pair<int, string> &p1;
    p1= p;
    p1.first= 7;
    cout<< "Value of P " << p.first << " " << p.second<< endl;
    cout<< "Value of P1 " << p1.first << " " << p1.second<< endl;
}
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL C/C++ Compile Run

```
[stl.cpp 2022-03-28 06:50:31.905]
,,stl.cpp: In function 'int main()':
stl.cpp:15:24: error: 'p1' declared as reference but not initialized
    pair<int, string> &p1;
                       ^~
```

Error compiling!

28 March 2022 12:21

How to debug the last problem?

```
pair<int, string> &p1;
p1= p;
```

What changes will you make here in order to make it work?

```
14
15      pair<int, string> &p1;
16      p1= p;
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL

```
[stl.cpp 2022-03-28 06:52:51.505]
,,stl.cpp: In function 'int main()':
stl.cpp:15:24: error: 'p1' declared as reference but not initialized
     pair<int, string> &p1;
                        ^~
```

28 March 2022 12:23

```
int main()
{
    pair<int, string> p;
    // p = make_pair(2, "abc"); //Using inbuilt function to
                                //add element in the pair

    p = { 2, "ankit"};
    // Now we can also copy the pairs just like variable

    pair<int, string> &p1= p;
    p1.first= 7;
    cout<< "Value of P " << p.first << " " << p.second<< endl;
    cout<< "Value of P1 " << p1.first << " " << p1.second<< endl;
}
```

C:\Windows\system32\cmd.exe

```
Value of P 7 ankit
Value of P1 7 ankit

Press any key to continue . . .
```

So just like variables, over here we are making use of reference to make changes in the original pair.

28 March 2022 12:25

But Why did we use pair?

- It is used to maintain the relationship between two things.

```
int main()
{
    int height[3]={4, 3, 5};
    int weight[3]={60, 50, 65};

    // Let's say we have two arrays,
    // height array contains respective height of student
    // weight array contains respective weight of student

    // for i th student,
    // height and weight would be height[i] & weight[i]
    // respectively

    //Now we have to create relationship between each ith
    //element by using pair

    // so if we want to swap data of 1st and 3rd student, the
    // swapping should happen in both the array,

    //creating array of pairs
    pair<int, int> p_array[3];
    for(int i=0;i<3;i++)
    {
        p_array[i]={height[i], weight[i]};
    }

    //Let's print the pairs

    for(int i=0;i<3;i++)
    {
        cout<< p_array[i].first << " " << p_array[i].second <<endl;
    }

    swap(p_array[0], p_array[2]);
    cout<<endl;
    for(int i=0;i<3;i++)
    {
        cout<< p_array[i].first << " " << p_array[i].second <<endl;
    }
    //here the relationship is maintained by itself
    //generally we declare vector of pair


}
```

```
int main()
```

```
{
    int height[3]={4, 3, 5};
    int weight[3]={60, 50, 65};
    // Let's say we have two arrays,
    // height array contains respective height of
student
    // weight array contains respective weight of
student

    // for i th student,
    // height and weight would be height[i] &
weight[i]
    // respectively
    //Now we have to create relationship between each
ith
    //element by using pair
    // so if we want to swap data of 1st and 3rd
student, the
    // swapping should happen in both the array,
    //creating array of pairs
    pair<int, int> p_array[3];
    for(int i=0;i<3;i++)
    {
        p_array[i]={height[i], weight[i]};
    }
    //Let's print the pairs
    for(int i=0;i<3;i++)
    {
        cout<< p_array[i].first << " "
<< p_array[i].second <<endl;
    }
    swap(p_array[0], p_array[2]);
    cout<<endl;
    for(int i=0;i<3;i++)
    {
        cout<< p_array[i].first << " "
<< p_array[i].second <<endl;
    }
    //here the relationship is maintained by itself
    //generally we declare vector of pair


}
```

We don't usually work with pair of arrays, we will be knowing more about it after we learn Vectors.

28 March 2022 12:59

Taking input in case of pairs

```
int main()
{
    pair<int, string> p;
    cout<<"Enter the first"<<endl;
    cin>> p.first;
    cout<< p.first << " " << p.second<<endl;
    cout<<"Enter the second"<<endl;
    cin>>p.second;
    cout<< p.first << " " << p.second<<endl;

}
```

C:\Windows\system32\cmd.exe

```
Enter the first
5
5
Enter the second
6
5 6

Press any key to continue . . .
```


```
int main()
{
    pair<int, string> p;
    cout<<"Enter the first"<<endl;
    cin>> p.first;
    cout<< p.first << " " << p.second<<endl;
    cout<<"Enter the second"<<endl;
    cin>>p.second;
    cout<< p.first << " " << p.second<<endl;
}
```

28 March 2022 12:59

Vectors:

They are very similar to arrays.
- They are also contiguous memory blocks.
- They are of dynamic size , not static like arrays.

Declaring Vectors:

```
int main()
{
    int arr[10]; //declared the chunch of 10 blocks
    vector< int> vec; //declaring vector of 0 size;

    // we can make vectors of any data type, and not only that
    // we can also make vectors of other containers.
    vector< pair<int, int>> j;

}
```


```
int main()
{
    int arr[10]; //declared the chunch of 10 blocks
    vector< int> vec; //declaring vector of 0 size;
    // we can make vectors of any data type, and not
only that
    // we can also make vectors of other containers.
    vector< pair<int, int>> j;
}
```

28 March 2022 13:52

Taking input in the vector:

```
int main()
{
    vector< int> vec;
    vec.push_back(10); //time complexity: O(1)
    vec.push_back(12);
    vec.push_back(4);
    vec.push_back(7);
}
```


```
int main()
{
    vector< int> vec;
    vec.push_back(10); //time complexity: O(1)
    vec.push_back(12);
    vec.push_back(4);
    vec.push_back(7);
}
```

28 March 2022 13:54

Size of the vector:

As the size of the vector is dynamic, so there is a function to help us get the size of the vector.
Time complexity is O(1);

```
int main()
{
    vector< int> vec;
    vec.push_back(10); //time complexity: O(1)
    vec.push_back(12);
    vec.push_back(4);
    vec.push_back(7);

    int size=vec.size();
    cout<<"Current size is "<< size<< endl;
    vec.push_back(9);
    size=vec.size();
    cout<<"After adding 9, size becomes "<<size<<endl;

}
```

C:\Windows\system32\cmd.exe

```
Current size is 4
After adding 9, size becomes 5

Press any key to continue . . .
```

```
int main()
{
    vector< int> vec;
    vec.push_back(10); //time complexity: O(1)
    vec.push_back(12);
    vec.push_back(4);
    vec.push_back(7);
    int size=vec.size();
    cout<<"Current size is "<< size<< endl;
    vec.push_back(9);
    size=vec.size();
```

```
    cout<<"After adding 9, size becomes "<<size<<endl;

}
```

28 March 2022 13:57

Printing the values stored in the vector:

Time complexity of size() is O(1).

```
int main()
{
    vector< int> vec;
    vec.push_back(10); //time complexity: O(1)
    vec.push_back(12);
    vec.push_back(4);
    vec.push_back(7);
    vec.push_back(9);

    cout<<"Size of vector is = "<<vec.size()<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i] << " ";
    }
}
```

C:\Windows\system32\cmd.exe

```
Size of vector is = 5
10 12 4 7 9
Press any key to continue . . .
```

```
int main()
{
    vector< int> vec;
    vec.push_back(10); //time complexity: O(1)
    vec.push_back(12);
    vec.push_back(4);
    vec.push_back(7);
    vec.push_back(9);
    cout<<"Size of vector is = "<<vec.size()<<endl;
    for(int i=0; i<vec.size();i++)
    {
```

```
        cout<< vec[i] << " ";
    }
}
```

28 March 2022 14:04

One thing to note:

When you declare the arrays locally in the function, you can't declare size more than 10^5 and if globally, the not more than 10^7
Same limits applies to vectors as well.

Just to avoid runtime error on online programming judges.

This limit is on contiguous memory allocation, so it applies to both arrays and vectors.

28 March 2022 14:22

We can also declare vector with the given size:
Let's say size is 7

```
void printVector(vector<int>vec)
{

    for(int i=0; i<vec.size();i++)…
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i] << "    ";
    }
}
int main()
{
    vector< int> vec(7);
    printVector(vec);

}
```

C:\Windows\system32\cmd.exe

```
0    1    2    3    4    5    6
0    0    0    0    0    0    0
Press any key to continue . . .
```

```
void printVector(vector<int>vec)
{
    for(int i=0; i<vec.size();i++)
    {
        cout<< i << "    ";
    }
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i] << "    ";
```

```
    }
}
int main()
{
    vector< int> vec(7);
    printVector(vec);
}
```

It will be declared of size 7 but it doesn't mean that it can't be expanded. I mean it doesn't mean that we can't add more no. of elements to this vector.

Let's see an example.

28 March 2022 14:27

After we pushed back "5", the size of the vector increased to 8 in order to accommodate this.

```
void printVector(vector<int>vec)
{

    for(int i=0; i<vec.size();i++) ...
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i] << "    ";
    }
}
int main()
{
    vector< int> vec(7);
    vec.push_back(5);
    printVector(vec);

}
```

C:\Windows\system32\cmd.exe

```
0    1    2    3    4    5    6    7
0    0    0    0    0    0    0    5
Press any key to continue . . .
```

```
void printVector(vector<int>vec)
{
    for(int i=0; i<vec.size();i++)
    {
        cout<< i << "    ";
    }
```

```
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i] << "    ";
    }
}
int main()
{
    vector< int> vec(7);
    vec.push_back(5);
    printVector(vec);
}
```

28 March 2022 14:29

Not only declaring with a specific size, we can also specify the values which has to be pre-filled with,

```
void printVector(vector<int>vec)
{

    for(int i=0; i<vec.size();i++)…
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i] << "    ";
    }
}
int main()
{
    vector< int> vec(7, 4);
    //(size, values to be prefilled)
    printVector(vec);

}
```

28 March 2022 14:31

Pop back function usage:

Both push_back and pop_back works with TC of O(1);

```
int main()
{
    vector< int> vec(7, 4);
    //(size, values to be prefilled)
    printVector(vec);
    cout<<endl;
    vec.pop_back();
    printVector(vec);

}
```

C:\Windows\system32\cmd.exe

```
0    1    2    3    4    5    6
4    4    4    4    4    4    4
0    1    2    3    4    5
4    4    4    4    4    4
Press any key to continue . . .
```


```
int main()
{
    vector< int> vec(7, 4);
    //(size, values to be prefilled)
    printVector(vec);
    cout<<endl;
    vec.pop_back();
    printVector(vec);
}
```

28 March 2022 14:33

Copying vector directly.

We can't directly copy arrays but we can do that in case of vectors.

In vectors we can create another vector with same value.

V2 is the copy of v

This is an expensive operation.

```
int main()
{
    vector< int> vec(7, 4);
    //(size, values to be prefilled)
    printVector(vec);
    cout<<endl;
    vec.pop_back();
    vector<int> vec2=vec; // O(n) TC in copying vector
    printVector(vec);
    cout<<endl;
    vec2.push_back(7);
    cout<<"Printing vec2: "<<endl;
    printVector(vec2);

}
```

C:\Windows\system32\cmd.exe

```
0    1    2    3    4    5    6
4    4    4    4    4    4    4
0    1    2    3    4    5
4    4    4    4    4    4
Printing vec2:
0    1    2    3    4    5    6
4    4    4    4    4    4    7
Press any key to continue . . .
```

28 March 2022 15:01

One very important thing in case of vectors.

When we are passing vector in printVector() function, then a local copy of this vector is created in printVector() function which is an expensive operation.

This operation is expensive.
Because a copy is being done which takes O(n) time.

```
void printVector(vector<int>vec)
{

    for(int i=0; i<vec.size();i++)···
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i] << "    ";
    }
}
int main()
{
    vector< int> vec(7, 4);
    //(size, values to be prefilled)
    printVector(vec);
}
```

C:\Windows\system32\cmd.exe

```
    1    2    3    4    5    6
    4    4    4    4    4    4
ress any key to continue . . .
```

28 March 2022 15:03

So to avoid the expense of copying, we can do the same operation by passing reference.

Note: We have used vector<int> &vec instead of vector<int> vec to avoid the expense of copying.

```
void printVector(vector<int> &vec)
{

    for(int i=0; i<vec.size();i++)···
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i] << "    ";
    }
    vec.push_back(6);
    cout<<endl;
}
int main()
{
    vector< int> vec(7, 4);
    //(size, values to be prefilled)
    printVector(vec);
    printVector(vec);

}
```

C:\Windows\system32\cmd.exe

```
0    1    2    3    4    5    6
4    4    4    4    4    4    4
0    1    2    3    4    5    6    7
4    4    4    4    4    4    4    6

Press any key to continue . . .
```

28 March 2022 15:07

Here we have made changes in "vec1" but it is being reflected in "vec" as well.

```
void printVector(vector<int> &vec)
{

    for(int i=0; i<vec.size();i++)…
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i] << "    ";
    }
    cout<<endl;
}
int main()
{
    vector< int> vec(7, 4);
    //(size, values to be prefilled)
    printVector(vec);
    vector< int> &vec1= vec;
    printVector(vec1);
    vec1.push_back(7);
    printVector(vec);
}
```

C:\Windows\system32\cmd.exe

```
0    1    2    3    4    5    6
4    4    4    4    4    4    4
0    1    2    3    4    5    6
4    4    4    4    4    4    4
0    1    2    3    4    5    6    7
4    4    4    4    4    4    4    7

Press any key to continue . . .
```

```
void printVector(vector<int> &vec)
{
    for(int i=0; i<vec.size();i++)
    {
        cout<< i << "    ";
```

```
    }
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i] << "    ";
    }
    cout<<endl;
}
int main()
{
    vector< int> vec(7, 4);
    //(size, values to be prefilled)
    printVector(vec);
    vector< int> &vec1= vec;
    printVector(vec1);
    vec1.push_back(7);
    printVector(vec);
}
```

28 March 2022 15:08

Let's make the vector of strings:

```
void printVector(vector<string> &vec)
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i] << "    ";
    }
    cout<<endl;
}
int main()
{
    vector<string> vec;
    int n;
    cout<<"Enter number of strings: ";
    cin>>n;
    for(int i=0;i<n;i++)
    {
        string s;
        cout<<endl<<"Enter" << i<< " string:";
        cin>>s;
        vec.push_back(s);
    }
    printVector(vec);

}
```

C:\Windows\system32\cmd.exe

```
Enter number of strings: 4

Enter0 string:Raka

Enter1 string:gyan

Enter2 string:vibhu

Enter3 string:Ank

Raka    gyan    vibhu    Ank

Press any key to continue . . .
```

28 March 2022 16:18

Not only for string or other data types, we can also make vectors of other containers, vector of vector which we will see soon!!

28 March 2022 16:21

Vector of pairs

Declaring vector of pairs

```
int main()
{
    vector< pair<int,int> > vec;
    // Every element of this vector is going to be
    // a pair with two values which are related to each other.
}
```

We can also initialize it over here itself.

```
void printVector(vector<pair<int,int>> &vec)
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i].first << ","<< vec[i].second<< endl;
    }
    cout<<endl;
}
int main()
{
    vector< pair<int,int> > vec;
    // Every element of this vector is going to be
    // a pair with two values which are related to each other.

    vec= {{1,1}, {8,9}, {5,4}, {7, 7}};
    printVector(vec);
}
```

There are other scary way of initialization, which we should be knowing.

```
void printVector(vector<pair<int,int>> &vec)
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i].first << ","<< vec[i].second<< endl;
    }
    cout<<endl;
}
int main()
{
    vector< pair<int,int> > vec= {{1,1}, {8,9}, {5,4}, {7, 7}};
    // Every element of this vector is going to be
    // a pair with two values which are related to each other.
    printVector(vec);
}
```

C:\Windows\system32\cmd.exe

```
1,1
8,9
5,4
7,7
```

```
void printVector(vector<pair<int,int>> &vec)
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i].first
<< ","<< vec[i].second<< endl;
    }
    cout<<endl;
}
int main()
{
    vector< pair<int,int> > vec= {{1,1}, {8,9}, {5,4},
{7, 7}};
    // Every element of this vector is going to be
    // a pair with two values which are related to
each other.
```

```
    printVector(vec);
}
```

28 March 2022 16:27

Taking input from the user for Vector of pairs.

There are two ways to push back:

- Here we are making use of vec.push_back({first, second});
- Another is to do vec.push_back(make_pair(first, second));

```
void printVector(vector<pair<int,int>> &vec)
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i].first << ","<< vec[i].second<< endl;
    }
    cout<<endl;
}
int main()
{
    vector< pair<int,int> > vec;
    int n;
    cout<<"No of elements: ";
    cin>>n;
    cout<<endl;
    for(int i=0;i<n;i++)
    {
        int first, second;
        cin>>first >> second;
        vec.push_back({first, second});
    }
    printVector(vec);
}
```

C:\Windows\system32\cmd.exe

```
No of elements: 4

1 2
6 8
9 8
7 5

1,2
6,8
9,8
7,5
```

```
void printVector(vector<pair<int,int>> &vec)
```

```
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i].first
<< ","<< vec[i].second<< endl;
    }
    cout<<endl;
}
int main()
{
    vector< pair<int,int> > vec;
    int n;
    cout<<"No of elements: ";
    cin>>n;
    cout<<endl;
    for(int i=0;i<n;i++)
    {
        int first, second;
        cin>>first >> second;
        vec.push_back({first, second});
    }
    printVector(vec);
}
```

28 March 2022 17:25

Second method with vec.push_back(make_pair(first, second));

```
void printVector(vector<pair<int,int>> &vec)
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i].first << ","<< vec[i].second<< endl;
    }
    cout<<endl;
}
int main()
{
    vector< pair<int,int> > vec;
    int n;
    cout<<"No of elements: ";
    cin>>n;
    cout<<endl;
    for(int i=0;i<n;i++)
    {
        int first, second;
        cin>>first >> second;
        vec.push_back(make_pair(first, second));
    }
    printVector(vec);
}
```

C:\Windows\system32\cmd.exe

```
No of elements: 3

1 3
6 4
9 2

1,3
6,4
9,2
```

```
void printVector(vector<pair<int,int>> &vec)
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
```

```
    {
        cout<< vec[i].first
<< ","<< vec[i].second<< endl;
    }
    cout<<endl;
}
int main()
{
    vector< pair<int,int> > vec;
    int n;
    cout<<"No of elements: ";
    cin>>n;
    cout<<endl;
    for(int i=0;i<n;i++)
    {
        int first, second;
        cin>>first >> second;
        vec.push_back(make_pair(first, second));
    }
    printVector(vec);
}
```

28 March 2022 17:25

Array of vectors.

How to declare?

```
int main()
{
    vector<int> v[10];
    // so this has made 10 vectors of zero size each,
    //It means that v[0], v[1], v[2] and so on are all individual vectors.

}
```


```
int main()
{
    vector<int> v[10];
    // so this has made 10 vectors of zero size each,
    //It means that v[0], v[1], v[2] and so on are all individual vectors.

}
```

28 March 2022 17:46

How to push values inside array of vectors?

```
void printVector(vector<int> &vec)
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i]<< endl;
    }
}
int main()
{
    int size;
    cout<<"size of vector array:";
    cin>>size;
    vector<int> vec[size];
    cout<<endl;
    for(int i=0;i<size;i++)
    {
        int size_of_ith_vector;
        cout<<"Enter size of "<< i<<"th vector: ";
        cin>>size_of_ith_vector;
        cout<<endl;
        for(int j=0;j<size_of_ith_vector;j++)
        {
            int value_inside_vector;
            cout<<"Push what?: ";
            cin>>value_inside_vector;
            vec[i].push_back(value_inside_vector);
        }
    }
    for(int i=0;i<size;i++)
    {
        printVector(vec[i]);
    }

}
```

C:\Windows\system32\cmd.exe

```
Enter size of 0th vector: 4

Push what?: 1
Push what?: 5
Push what?: 6
Push what?: 9
Enter size of 1th vector: 2

Push what?: 6
Push what?: 5
Enter size of 2th vector: 3

Push what?: 8
Push what?: 7
Push what?: 9

1
5
6
9

6
5

8
7
9
```

```
void printVector(vector<int> &vec)
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
```

```
    {
        cout<< vec[i]<< endl;
    }
}
int main()
{
    int size;
    cout<<"size of vector array:";
    cin>>size;
    vector<int> vec[size];
    cout<<endl;
    for(int i=0;i<size;i++)
    {
        int size_of_ith_vector;
        cout<<"Enter size of "<< i<<"th vector: ";
        cin>>size_of_ith_vector;
        cout<<endl;
        for(int j=0;j<size_of_ith_vector;j++)
        {
            int value_inside_vector;
            cout<<"Push what?: ";
            cin>>value_inside_vector;
            vec[i].push_back(value_inside_vector);
        }
    }
    for(int i=0;i<size;i++)
    {
        printVector(vec[i]);
    }

}
```

28 March 2022 17:46

Vectors of Vectors

- These are very very useful
- You can make the size of the array dynamic here as well.

Array of array is a 2d array where
-No of rows and columns are fixed

Array of vectors behave like 2d array where
-No of rows are fixed.
-No of columns can be changed dynamically.

```
void printVector(vector<int> &vec)
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i]<< " ";
    }
}
int main()
{
    int size;
    cout<<"size of vector array:";
    cin>>size;
    vector<int> vec[size];
    cout<<endl;
    for(int i=0;i<size;i++)
    {
        int size_of_ith_vector;
        cout<<"Enter size of "<< i<<"th vector: ";
        cin>>size_of_ith_vector;
        cout<<endl;
        for(int j=0;j<size_of_ith_vector;j++)
        {
            int value_inside_vector;
            cout<<"Push what?: ";
            cin>>value_inside_vector;
            vec[i].push_back(value_inside_vector);
        }
    }
    for(int i=0;i<size;i++)
    printVector(vec[i]);
    cout<<endl<<"value at 0th row and 1st column"<< vec[0][1];

}
```

C:\Windows\system32\cmd.exe

```
size of vector array:2

Enter size of 0th vector: 3

Push what?: 1
Push what?: 6
Push what?: 5
Enter size of 1th vector: 2

Push what?: 8
Push what?: 7

1 6 5
8 7
value at 0th row and 1st column6
Press any key to continue . . .
```

```
void printVector(vector<int> &vec)
{
    cout<<endl;
    for(int i=0; i<vec.size();i++)
    {
        cout<< vec[i]<< " ";
    }
}
int main()
{
    int size;
    cout<<"size of vector array:";
    cin>>size;
    vector<int> vec[size];
    cout<<endl;
    for(int i=0;i<size;i++)
    {
        int size_of_ith_vector;
        cout<<"Enter size of "<< i<<"th vector:
";
        cin>>size_of_ith_vector;
        cout<<endl;
        for(int j=0;j<size_of_ith_vector;j++)
        {
            int value_inside_vector;
            cout<<"Push what?: ";
            cin>>value_inside_vector;
            vec[i].push_back(value_inside_vecto
r);
        }
    }
    for(int i=0;i<size;i++)
    printVector(vec[i]);
    cout<<endl<<"value at 0th row and 1st
column"<< vec[0][1];

}
```

28 March 2022 17:51

But in case of **vectors of vectors,**
Both the number rows and columns can be made dynamic.

How to declare this?