Concept of Vectors:

Vectors are same as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted, with their storage being handled automatically by the container.

Vector elements are placed in contiguous storage so that they can be accessed and traversed using iterators. In vectors, data is inserted at the end.

Inserting at the end takes differential time, as sometimes there may be a need of extending the array.

Removing the last element takes only constant time because no resizing happens.

Inserting and erasing at the beginning or in the middle is linear in time.

Commonly used vector Functions:

1) **begin()** – Returns an iterator pointing to the first element in the vector

2) **end()** – Returns an iterator pointing to the theoretical element that follows the last element in the vector

3) **size()** – Returns the number of elements in the vector.

4) **reserve()** – For reversing an array Eg. (a.begin(), a.end());

5) **empty()** – Returns whether the container is empty.

6) **reference operator [g]** – Returns a reference to the element at position 'g' in the vector

7) **at(g)** – Returns a reference to the element at position 'g' in the vector

8) **front()** – Returns a reference to the first element in the vector

9) **back()** – Returns a reference to the last element in the vector

10) **assign()** – It assigns new value to the vector elements by replacing old ones

11) **push_back()** – It push the elements into a vector from the back

12) **pop_back()** – It is used to pop or remove elements from a vector from the back.

13) **insert()** – It inserts new elements before the element at the specified position

14) **erase()** – It is used to remove elements from a container from the specified position or range.

Rapid five:

1) What do vectors represent? Answer: **Dynamic arrays**

2) In which type of storage location are the vector members stored? Answer: **Contiguous storage locations**

3) How many vector container properties are there in c++? Answer: **Three**

There are three container properties in c++. They are sequence, Dynamic array and allocator-aware.

4) Which is optional in the declaration of vector? Answer: **Number_of_elements**

5) Pick out the correct statement about vector.

a) vector<int> values (5)

b) vector values (5)

c) vector<int> (5)

d) vector<5>

How to declare a Vector //Header file to be used #include <vector> else prefer #include<bits/stdc++.h>

```
vector <int> s;
```

Code Fragments for performing Basic Operations

1) **s.push_back(element name); //Its used to insert element at back of the vector**

```
vector <int> insert_last(vector <int> s, int n)
{
        s.push_back(n);
        return s;
}
```

2) **s.pop_back(); //Its used to delete element at back of the vector**

```
vector <int> delete_last(vector <int> s)
{
        s.pop_back();
        return s;
}
```

3) **s.insert(s.begin, val) // Inserting in the beginning**

```
vector <int> insert_beg(vector <int> s, int n)
{
        s.insert(s.begin(),n);
        return s;
}
```

4) s.erase(s.begin()) // Deleting from the beginning

```
vector <int> delete_beg(vector <int> s)
{
        s.erase(s.begin());
        return s;
}
```

5) Printing Middle Element

```
void print_mid(vector <int> s)
{
        int n;
        n=s.size();
        if(n%2!=0)
```

```
        cout<<s[n/2];
        else
        cout<<s[n/2-1]; //for printing 1st mid use s[n/2] for 2nd mid
}
```

6. Deleting the middle element

```
vector <int> delete_mid(vector <int> s)
{
    int n=s.size();
    int a;
    a=n/2;
    if(n%2!=0)
        s.erase(s.begin()+a);
        else
        s.erase(s.begin()+a-1); //for deleting 1stt mid, to delete second mid use s.begin+a
        return s;
}
```

7. Inserting at middle position

```
vector <int> insert_mid(vector <int> s, int m)
{
        int n=s.size();
   int a;
   a=n/2;
   if(n%2!=0)
        s.insert(s.begin()+a,m);
        else
        s.insert(s.begin()+a-1,m);
        return s;
}
```

8. Printing the created vector

```
void print_vector(vector <int> s)
{
        int n;
        n=s.size();
        for(int i=0; i<n; i++)
```

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

Coding Practice:

1) Create a vector. Try adding 5 elements in it and then print all 5 elements.

2) Create a function to Insert and Delete (at below mention positions)

i) At beginning

ii) In middle

iii) At end

iv) At particular point

3) Write a program to Print Middle element of the vector.

4) Write a program to delete even numbers present in the list of vectors. (Hint use pop_back function)

Eg: Initial Element: 6

Input: 1 2 3 4 5 6

Output 1 3 5

5) Write a program to sort a vector. (In both Ascending and Descending order)

Hint:-

sort(s.begin(), s.end()); //Ascending order

sort(s.begin(), s.end(),greater<int>()); //Descending Order

6) Write a program to Quickly check if two STL vectors contain same elements or not.