```
          ceil(2.3);=3,ceil(-2.3)=-2 big int value
            floor(2.3);=2 ,floor(-2.3)=-3 small int value

        1.  BITWISE NOT OPERATOR=>
                                        ~N=-(N+1)
//N MUST BE IN BINARY FORM

   // NEGATIVE VALUE ALWAYS STORED IN 2's COMPLIMENT
          2. sorting=>

                    bool cmp(pair<int,int>p1,pair<int,int>p2){  //
(obj1,obj2);
                      int x=p1.first;
                      int y=p2.first;
                      if(x<y)  // agar x ,1 se jyada bar ayega to fcfs algo
                      return true;
                      else
                      return false;
                      }
                    sort(v.begin(),v.end(),cmp); //v={{2,3},{5,2},{2,4}};

                      int a[n];
                      sort(a,a+n);
                      sort(v.begin(),v.end());
                      sort(v.rbegin(),v.rend());
//sort(v.begin(),v.end(),greater<int>());
              2D vector sorting wrt  row->
                      sort(v[i].begin(),v[i].end());  //increasing
                      sort(v[i].rbegin(),v[i].rend()); //decreasing
            2D vector sorting wrt  colom->
                      bool sortcol( const vector<int>& v1, const vector<int>&
v2 ) {
                      return v1[i] <v2[i]; //incresing wrt i index colom
                      }
                      sort(v.begin(), v.end(),sortcol);

          2.1  2D function calling vector =>
                                  vector<vector<int>>
dp(n,vector<int>(w,-1)); //2D matrix with size of[n][w] all value is -1.
                                  vector<vector<int>> &dp;

          2.2      string numstr=to_string(cnt);   // change number to string
          2.3       string firstString = "Scaler";
                  string secondString = "Scaled";

              int equal = s1.compare(s2); // finding lexographically compare
                  equal=0  // s1==s2
                  equal>0  // s1>s2
                  equal<0  // s1<s2
```

```
                      int num = stoi(str);



              3.    Power =>
std::string bin("1001");
                   sum=pow(2 , x);                                    int
dec = std::stoi(bin, nullptr, 2); //string to any number system

              4.   In factorial forwarding call(recursive call) and then back
tracking occure( return 0, return 1) return 1 gives
                   value 1 to called function so backtracking multiplication do
not gives 0;

              5.   bool present=binary_search(v.begin(),v.end(),key);
                       int
freq=upper_bound(v.begin(),v.end(),key)-lower_bound(v.begin(),v.end(),key);
                   int lb=lower_bound(v.begin(),v.end(),key)-v.begin();
                    lb=>first occurence of  val equal or just greater
                    up=>just greater


              6.    HOW to take string input with space=>
                     std::string s;
                     std::getline(std::cin >> std::ws, s);

                    How to take input when number of test case NOT given->
                    while(cin>>x){..........................};

              7.   vector<int/bool,char>v;    dynamic memory
                    vector<int>v(5,10);   fixed size 10 with value 5 initialisd

                      vector<vector<int>>v;
                      for(int i=0;i<v.size();i++){
                          for(int j=0;j<v[i].size();j++
                           cout<<v[i][j]<<" ";
                           cout<<endl;
                      }
              7.    struct node {
                                        int data;
                                        node *left;
                                        node *right;
                                     node(int val){
                                     data=val;
                                     left=NULL;
                                     right=NULL;
                                     }
                      };
                    node *root=new node(23);
```

```
8.    <space> = 32  ,  [0-9]=[48-57]  ,[A-Z]=[65-90]   , [a-z]=[97-122]
//ascii charecter with decimal value
9.          int a=abs(-12);  //return absolute value of number or
expression
10.  __gcd(8,12);
11.  ios::sync_with_stdio(0);
12. auto keyword is used for any data declaration

13. map/multimap>       //stored data in sorder order w.r.t key
              int n,a;cin>>n;
             map<int, int> fre;int x=1;
               for(int i = 0; i < n; ++i){
                         cin >> a;
                         fre[a]++;        //by defalt value 0 hoti h
                         if(x<fre[a])
                         x=fre[a];
                 }

14.
              int n,input;cin>>n;
             set<int>st;    //store data in sorted order
            for(int i=0;i<n;i++){
                cin>>input;
               st.insert(input);
                 }
           //st.erase(100);st.count(200);
             for(auto i:st)
           cout<<i<<" ";

15.   queue<int>q;
             q.push(23);
             q.push(200);
             q.push(400);
          queue<int>temp=q;
            temp.front(); // 23
            temp.back(); //400
            temp.pop(); // remove 23 from front side
            while (!temp.empty()) {
       cout << temp.front() << " ";
             temp.pop();
   }

            priority_queue<int>pq;  //max heap
            priority_queue <int, vector<int>, greater<int> > pq;    //min
heap
        pq.push(5);
           pq.push(1);
           pq.push(10);
           pq.push(30);
```

```cpp
//   One by one extract items from min heap
while (!pq.empty()) {
        cout << pq.top() << " ";
        pq.pop();
}

map<string ,int>mp;    //store data in sorted order wrt key
mp["vikas"]=30;
bool present=mp.count(key);
for(auto i:mp){
cout<<i.first<<" "<<i.second<<endl;


pair<string,int>p1;
p1=make_pair("vikas", 56);
p1.first<<p1.second<<endl;
cout<<get<0>(p1)<<"  "<<get<1>(p1)<<endl;
// tuple se acha to struct ka usedefined data type bna lo
tuple<string,int,int,bool>t1;
t1=make_tuple("vikas", 56,27,true);
 int n=tuple_size<decltype(t)>::value;
cout<<get<0>(t1)<<endl;
cout<<get<1>(t1)<<endl;

  Graph->
        directed->undirected
        weighted->unweighted
        cyclic ->acyclic
        joint->disjoint

        vector<int>adj[n];  Array of integer_vector     ->   data structure to
store the unweighted graph
        vector<pair<int,int>>adj[n];  Array of pair_vector ->  data structure to
store the weighted graph

        cin>>n>>m;    //n nodes and m edges
      vector<int> adj[n]; // 0 based indexing ,array of vector
    for(int i=0;i<m;i++){
    int u,v; cin>>u>>v;
    adj[u].push_back(v);
    adj[v].push_back(u);   }
```