```
#include <iostream>
using namespace std;
void exch(int a[],int i,int j){
    int s=a[i];
    a[i]=a[j];
    a[j]=s;

}
int  partition(int a[],int l,int h);
void quick(int a[],int l,int h){
    if (h<=l) return ;
    int j=partition(a,l,h);
    quick(a,l,j-1);
    quick(a,j+1,h);
    }
int partition(int a[],int l,int h){
    int i=l-1;
    int j=h;
    int v=a[h];
    while(true){

        while( a[++i]<v);

        while(a[--j]>v) if (j==i)  break;

            if (i>=j) break;

        exch(a,i,j);

    }

    exch(a,i,h);
    return i;



}
int main(){

    int a[]={12,43,13,5,8,10,11,9,20,17};
    int n=sizeof(a)/sizeof(int);
quick(a,0,n-1);
 for (int  i=0;i<n;i++){
     cout<<a[i]<<"  ";
 }
     return 0;
 }



 //////////////////////////////////////////




 /*
 * STL Library Examples
 *
 * Author: Hu Yuhuang
 * Date  : 2014-09-14
 */

/*** System Library ***/
#include<iostream>
#include<cstdio>
#include<cmath>
#include<cstdlib> // random
```

```
#include<ctime>
#include<climits> // all useful constants
#include<functional> // for hash
#include<algorithm>
#include<sstream>

/*** Data Structure ***/
#include<string>
#include<queue>
#include<stack>
#include<vector>
#include<deque> // double ended queue
#include<list> // priority queue

using namespace std;

int max(int a, int b)
{
    return a>b ? a:b;
}

int min(int a, int b)
{
    return a<b ? a:b;
}

int gcd(int a, int b)
{
    if (b==0) return a;
    else return gcd(b, a%b);
}

int lcm(int a, int b)
{
    return a*b/gcd(a,b);
}

bool prime(int n)
{
    if (n<2) return false;
    for (int i=2;i*i<=n;i++)
        if (n%i==0) return false;
    return true;
}

bool isLeap(int n)
{
    if (n%100==0)
        if (n%400==0) return true;
        else return false;

    if (n%4==0) return true;
    else return false;
}

long powmod(long base, long exp, long modulus) {
  base %= modulus;
  long result = 1;
  while (exp > 0) {
    if (exp & 1) result = (result * base) % modulus;
    base = (base * base) % modulus;
    exp >>= 1;
  }
  return result;
}

int factmod (int n, int p) {
    long long res = 1;
    while (n > 1) {
        res = (res * powmod (p-1, n/p, p)) % p;
        for (int i=2; i<=n%p; ++i)
            res=(res*i) %p;
        n /= p;
    }
    return int (res % p);
}

void combination(int n, int m)
{
    if (n<m) return ;
    int a[50]={0};
    int k=0;

    for (int i=1;i<=m;i++) a[i]=i;
    while (true)
    {
        for (int i=1;i<=m;i++)
            cout << a[i] << " ";
        cout << endl;

        k=m;
        while ((k>0) && (n-a[k]==m-k)) k--;
        if (k==0) break;
        a[k]++;
        for (int i=k+1;i<=m;i++)
            a[i]=a[i-1]+1;
    }
}

int main(void)
{
    /**** Max or min ****/

    cout << "------TEST FOR MAX OR MIN------" << endl;
    // Max and min are implemented in library
    // the point for this is to show the syntax

    cout << "Max of (5,7): " << max(5,7) << endl;
    cout << "Min of (5,7): " << min(5,7) << endl;

    cout << "------TEST FOR MAX OR MIN------" << endl;
    cout << endl << endl;

    /**** GCD and LCM ****/

    cout << "------TEST FOR GCD AND LCM------" << endl;

    cout << "GCD of (12, 15): " << gcd(12,15) << endl;
    cout << "LCM of (12, 15): " << lcm(12,15) << endl;

    cout << "------TEST FOR GCD AND LCM------" << endl;
    cout << endl << endl;

    /**** prime number ****/

    cout << "------TEST FOR PRIME NUMBER------" << endl;

    cout << "Is 1251 prime?: " << prime(1251) << endl;
    cout << "Is 97 prime?  : " << prime(97) << endl;

    cout << "------TEST FOR PRIME NUMBER------" << endl;
    cout << endl << endl;

    /**** Leap year ****/
    cout << "------TEST FOR LEAP YEAR------" << endl;

    cout << "2012 is Leap? : " << isLeap(2012) << endl;
    cout << "1900 is Leap? : " << isLeap(1900) << endl;
    cout << "1903 is Leap? : " << isLeap(1903) << endl;

    cout << "------TEST FOR LEAP YEAR------" << endl;
    cout << endl << endl;

    /**** a^b mod p and n! mod p ****/

    cout << "------TEST FOR (A^B MOD P) AND (N! MOD P)------" << endl;

    cout << "5^6 mod 17: " << powmod(5, 6, 17) << endl;
    cout << "17 mod 17 : " << factmod(17, 17) << endl;

    cout << "------TEST FOR (A^B MOD P) AND (N! MOD P)------" << endl;
    cout << endl << endl;

    /**** Generate combinations ****/

    cout << "------TEST FOR GENERATING COMBINATIONS------" << endl;

    combination(6, 3); // pick 3 numbers from 6 numbers

    cout << "------TEST FOR GENERATING COMBINATIONS------" << endl;
    cout << endl << endl;

    return 0;
}
```

```
 /*
 * STL Library Examples
 *
 * Author: Hu Yuhuang
 * Date  : 2014-09-14
 */

/*** System Library ***/
#include<iostream>
#include<cstdio>
#include<cmath>
#include<cstdlib> // random
#include<ctime>
#include<climits> // all useful constants
#include<functional> // for hash
#include<algorithm>
#include<sstream>

/*** Data Structure ***/
#include<string>
#include<queue>
#include<stack>
#include<vector>
#include<deque> // double ended queue
```

```
#include<list> // priority queue

#define int long long // make everyone long long
#define double long double;

using namespace std;

// example for condition
bool isOdd(int n)
{
    return ((n%2)==1);
}

// example for compare
bool small(int n, int m)
{
    return n<m;
}

#undef int // main must return int
int main(void)
#define int long long // redefine int
{
    /**** Limits ****/

    cout << "------TEST FOR LIMITS------" << endl;
    // All defined in <climits>
    cout << INT_MAX << endl;
    cout << INT_MIN << endl;
    cout << LONG_MAX << endl;
    cout << LONG_MIN << endl;
    cout << LLONG_MAX << endl;
    cout << LLONG_MIN << endl;
    cout << (~0u) << endl; // 4294967295 in my system
    cout << "------TEST FOR LIMITS------" << endl;
    cout << endl << endl;

    /**** Space Waster ****/
    // The following tests are carried out with space waster
    // unless it's mentioned.


    /**** Initialize array with predefined value ****/

    int a[10]={0}; // initialize every elements as 0
                   // working only for 1-d array
                   // if not use other numbers between {},
                   // first element will be the value, and rest
                   // are 0.

    cout << "------TEST FOR INITIALIZE ARRAY------" << endl;
    for (int i=0;i<10;i++)
        cout << a[i] << " ";
    cout << endl;

    fill(a, a+10, 2);
    for (int i=0;i<10;i++)
        cout << a[i] << " ";
    cout << endl;

    fill_n(a, 10, 3);
    for (int i=0;i<10;i++)
        cout << a[i] << " ";
    cout << endl;

    memset(a, 0, sizeof(a));
    for (int i=0;i<10;i++)
```

```
        cout << a[i] << " ";
    cout << endl;

    int b[10][10];
    for (int i=0;i<10;i++)
        fill(b[i], b[i]+10, 5);

    for (int i=0;i<10;i++)
    {
        for (int j=0;j<10;j++)
            cout << b[i][j] << " ";
        cout << endl;
    }
    cout << "------TEST FOR INITIALIZE ARRAY------" << endl;
    cout << endl << endl;

    /**** Modifying sequence operations ****/
    cout << "------TEST FOR MODIFYING SEQUENCE OPERATIONS------" << endl;

    // generate data
    for (int i=0;i<10;i++) a[i]=i;
    cout << "Original data:   ";
    for (int i=0;i<10;i++) cout << a[i] << " ";
    cout << endl;

    // copy
    int c[5];
    copy(a,a+5,c);
    cout << "Copied data:     ";
    for (int i=0;i<5;i++) cout << c[i] << " ";
    cout << endl;

    // swap
    swap(a[2], a[6]);
    cout << "Swapped data:    ";
    for (int i=0;i<10;i++) cout << a[i] << " ";
    cout << endl;

    // replace
    fill(a, a+10, 4);
    cout << "Original data:   ";
    for (int i=0;i<10;i++) cout << a[i] << " ";
    cout << endl;
    cout << "Replaced data:   ";
    replace(a,a+4, 4, 5);
    for (int i=0;i<10;i++) cout << a[i] << " ";
    cout << endl;

    // replace if
    for (int i=0;i<10;i++) a[i]=i;
    cout << "Original data:   ";
    for (int i=0;i<10;i++) cout << a[i] << " ";
    cout << endl;
    replace_if(a, a+10, isOdd, 0); // replace all odd number to 0
    cout << "Replace-if data: ";
    for (int i=0;i<10;i++) cout << a[i] << " ";
    cout << endl;

    // reverse
    reverse(a, a+7);
    cout << "Reversed data:   ";
    for (int i=0;i<10;i++) cout << a[i] << " ";
    cout << endl;

    // reverse-copy is similar to copy

    // random_shuffle
    for (int i=0;i<10;i++) a[i]=i;
    cout << "Original data:       ";
    for (int i=0;i<10;i++) cout << a[i] << " ";
    cout << endl;
    random_shuffle(a,a+10);
    cout << "Random shuffle data: ";
    for (int i=0;i<10;i++) cout << a[i] << " ";
    cout << endl;
    cout << "------TEST FOR MODIFYING SEQUENCE OPERATIONS------" << endl;
    cout << endl << endl;

    /**** Merge ****/
    cout << "------TEST FOR MERGE------" << endl;

    int d[10]={0};
    int e[10]={0};

    for (int i=0;i<10;i++) d[i]=i;
    cout << "Original data 1:       ";
    for (int i=0;i<10;i++) cout << d[i] << " ";
    cout << endl;
    for (int i=0;i<10;i++) e[i]=i+1;
    cout << "Original data 2:       ";
    for (int i=0;i<10;i++) cout << e[i] << " ";
    cout << endl;

    // merge
    int f[20]={0};
    merge(d, d+10, e, e+5, f, small);
    // same effect without the last term
    cout << "Merged data:           ";
    for (int i=0;i<15;i++) cout << f[i] << " ";
    cout << endl;

    // set union
    int g[20]={0};
    set_union(d, d+10, e, e+10, g);
    cout << "Set union data:        ";
    for (int i=0;i<20;i++) cout << g[i] << " ";
    cout << endl;

    // set intersection
    int h[20]={0};
    set_intersection(d, d+10, e, e+10, h);
    cout << "Set intersection data: ";
    for (int i=0;i<20;i++) cout << h[i] << " ";
    cout << endl;

    // set difference
    int l[20]={0};
    set_difference(d, d+10, e, e+10, l);
    cout << "Set difference data:   ";
    for (int i=0;i<20;i++) cout << l[i] << " ";
    cout << endl;

    cout << "------TEST FOR MERGE------" << endl;
    cout << endl << endl;

    /**** String ****/
    cout << "------TEST FOR STRING------" << endl;

    string st="abcldefghijklmn";
    cout << "String      : " << st << endl;
    cout << "Find first l: " << st.find("l") << endl;
    cout << "Find last l : " << st.rfind("l") << endl;
    cout << "Insert aaaa : " << st.insert(5, "aaaa") << endl;
    cout << "Erase aaaa  : " << st.erase(5, 4) << endl;
    cout << "Replace de  : " << st.replace(st.find("de"), 2, "ll") << endl;

    stringstream s1;
    int i=22;
    s1 << "Hello world" << i;
    cout << s1.str() << endl;

    cout << "------TEST FOR STRING------" << endl;
    cout << endl << endl;

    /**** Sort ****/
    cout << "------TEST FOR SORT------" << endl;
    for (int i=0;i<10;i++) a[i]=i;
    random_shuffle(a, a+10);
    cout << "Original data : ";
    for (int i=0;i<10;i++) cout << a[i] << " ";
    cout << endl;

    // sort
    sort(a, a+10);
    cout << "Sorted data   : ";
    for (int i=0;i<10;i++) cout << a[i] << " ";
    cout << endl;

    cout << "------TEST FOR SORT------" << endl;
    cout << endl << endl;

    /**** Permutations ****/
    cout << "------TEST FOR PERMUTATIONS" << endl;

    int o[4]={1,2,3,4};

    do
    {
        for (int i=0;i<4;i++) cout << o[i] << " ";
        cout << endl;
    }while (next_permutation(o, o+4));

    cout << "------TEST FOR PERMUTATIONS" << endl;
    cout << endl << endl;

    /**** Searching ****/
    // Similar to sring's search.

    /**** Random algorithm ****/

    cout << "------TEST FOR RANDOM ALGORITHM" << endl;

    srand(time(NULL));

    cout << "Rand number [5,10): " << rand()%(10-5)+5 << endl;
    cout << "Rand number [5,10]: " << rand()%(11-5)+5 << endl;

    cout << "------TEST FOR RANDOM ALGORITHM" << endl;
    cout << endl << endl;

    // use random_permuation like next_permuation

    return 0;
}
```