1.  You have a ladder n-steps in height.  You can either take one step

or two steps up the ladder at a time.  How can you find out all the

different combinations up the ladder?  Then figure out an algorithm that

will actually print out all the different ways up the ladder.  ie:

1,1,2,1,2,2... etc...

2.  Given the root node to a singly linked list, reverse the last 5

nodes in the list.  For a list with 5 or less nodes, reverse the whole list.

3.  Given the root node to a singly linked list, write an algorithm to

detect if there is a loop in the list.

4.  Write an algorithm to calculate the square root of a number.

5. Give an array of 100 random integers.  Write an algorithm to find the

the closest 2 integers (closest by position) in the array that add up to 100.

6. Given an array of integers, write a method that returns an array of the same size where each

index is the product of all integers except itself, ie given array {1,2,3,4} return {24,12,8,6}

explicitly {2\*3\*4,1\*3\*4,1\*2\*4,1\*2\*3}.

7.  Implement a deque.  Think about what properties a deque must have.

 Implement it as a base class that can be extended and as a template.

In general, start thinking about what's underneath all the other STL

data types and how they work.

8.  Given a char pointer to large buffer of memory, write your own

version of my\_malloc and my\_free without using any system calls.  Make

it as robust as possible.  How would you minimize memory fragmentation?

9. Giving a char array with only Xs and Ys, do an in-place separation of the Xs and Ys.

Example: "XYXXYYYXX" -> "YYYYXXXXX"

10.  Given the function: "bool numExists( int array[], int array\_len,

int num )" where "array" is a sorted array of integers.  Determine if

"num" exists in the array.

11. The standard library function of atoi() is not very robust.  How

would you design/implement a better version of it?

12. Implement a Singleton.  Everyone knows the "textbook" implementation of the

singleton but think of all the different ways you can implement it and

what are their pros/cons?  Make a thread-safe version.

Hi Ratikant,

Below is some information which can be useful for preparing for the Bloomberg interview.

HackerrankX is the website they use for interviewing, it is interactive which allows you to login as a candidate and code during the interview. So the interviewers can see your code. Hackerrank also has about 1000 questions which you can practice. [www.hackerrank.com](http://www.hackerrank.com/)

Bloomberg tests code writing capabilities which is very critical part of the BBG interview process. So you need to practice it extensively by actually writing code for small questions. A little research on some peculiar questions asked at BBG will help. Here is the link to some questions not all are helpful but they should try practicing some small questions to write code.  
  
<http://www.careercup.com/page?pid=bloomberg-lp-interview-questions&n=1>

**For quickly reviewing the STL containers for the interviews -**

<http://john-ahlgren.blogspot.com/2013/10/stl-container-performance.html>

for information on features please refer to [www.cplusplus.com](http://www.cplusplus.com/)

**Some Interview questions and code –**

1.

Void f (               )

{

          char \*p1 = “AMERICA”;

}

main()

{

          Char \*p = “HELLO”;

          //  f(   )

          printf(“ printing %s “, p );

}

Main as is will print HELLO. Uncomment   f() and add code in f()  in that way that printf in main prints AMERICA.

2.  Multithreaded server application stop working and last log message from the application is:

“some message from the server ...”

Code looks like:

Some F ()

{

Code ...

Asquiring Thread  lock

Line printing “some message from the server ...”

Func();

Relasing Thread Lock

}

What  programmer in charge will do in order to debug this?

What is happened wrong in the Func() ?

If an exception is thrown in the Func() what should be done to fix problem ?

3. What is “placement new” in C++

4.  What are internal of operator new in C++.  Is new just wrapper around malloc() ?

5. What is smart pointer . Write a SmartPtr class. What is difference between auto\_ptr and smart ptr?

6.  Pointers vs  references  used like a function argument

7.  They want to ask a C++ template question but skipped it.

8. Make a dictionary using  name  as a key and phone number  as a value for all humans in the world. What  data structure will be used if low access time is a  requirement? If the dictionary is used all the time in real time what auxiliary data structure will be used to handle erased ( dead) and new born in the dictionary?

         Difference between Static variables in C and C++?

         Difference between static and Dynamic Polymorphism?

         What is visitors pattern?

         When should you use virtual destructors?

         Write the code for reversing the link list

1.    Problem about trees – find one particular node in a tree, then do it iteratively

2.    Reversal in Word lists, single lists

3.    Maps, # maps, STL maps

4.    Shared pointers, auto pointers

5.    Singleton patterns, write singletons

6.    RAII

7.    Vectors of shared pointers

8.    Are Shared pointers thread safe or not?

Below is the code someone wrote on a scenario they were told to program –

// foo( int n)  
// prints all the prime numbers < n

 My full answer copied from interview GUI

 bool isPrime(int i)  
{   
    int j;  
    for ( j = 2; j< i; ++j)  // j=3; i=3  
    {  
        // if (i mod j != 0) implement it   
        int tmp = i / j;  // i 3 ; j 2    tmp = 1 tmp \* j= 1\*2 = 2  //// I added this line for my explaination for him   
        if ( i == (i / j)\*j)  
        {  
            return false;  
        }  
          
    }

    return true;  
}

void foo(int n)  
{  
    if (n < 2)  
       return;  
         
  //  int sqrt\_n = sqrt(n);  
      
    for (int i=2; i < n; ++i) //  prints all the prime numbers < n , no =  
    {  
        if (isPrime(i))  
           cout << i << " ";  
    }  
    cout << endl;  
}

// test case  
(1) 1  
(2) 2  
(3) -1  
(4)  -5  
(5) 7 (normal)

(6)  4

Another candidate was asked to find error in this code and was asked questions based on the following code –

The questions were based on the code below and touched upon   
virtual destructorsstatic variablesarray (new/delete)variable scope  
#include <otherclass.h>

class Base {  
  public:  
    virtual int get( int ii ) = 0;  
};

class Derived : public Base {  
  private:  
    int        \*array;  
    OtherClass \*object;

  public:  
    Derived(int n)  
    : array(new int[n]),  
      object(new OtherClass)  
    {  
    }  
    virtual int get( int ii )  
    {  
        return array[ii];  
    }  
};

===========================================================

Problem: Given a arrat of (repeating) numbers, print them in an order so that,

the least repeated digit comes first and most repeated digit comes in the last.

-------------------------------------------------------------

Solution 1: inplace sorting and printing (complexity: o( N log N) )

// input:           3,4,5,3,4,6,5,6,7,3,3

// expected output: 7,6,6,5,5,4,4,3,3,3,3

#include <iostream>

#include <algorithm>

#include <iterator>

using namespace std;

int main()

{

    int list[11] = {3,4,5,3,4,5,3,6,7,3,3};

    //int list[11] = {6,4,5,7,7,3,4,5,7,7,3};

    const int n = sizeof(list) / sizeof(int);

    sort(list, list+n, greater\_equal<int>());

    copy(list, list+n, ostream\_iterator<int>(cout, " "));

}

===========================================================

Solution 2: using a unordered multimap and ordered multimap ( complexity: o(N)

#include <iostream>

#include <map>

#include <unordered\_map>

using namespace std;

struct MostFreqFirst\_Compare

{

    bool operator() (int a, int b) { return a > b; }

};

int main()

{

      int list[11] = {3,4,5,3,4,5,3,6,7,3,3};

      //int list[11] = {6,4,5,7,7,3,4,5,7,7,3};

      unordered\_map<int, int> myumap;

      unordered\_map<int, int>::iterator itr;

      for( int i = 0 ; i < 11; i++)

      {

          itr = myumap.find(list[i]);

          if (itr != myumap.end())

              itr->second++;

          else

              myumap.insert(make\_pair<int, int>(list[i],0) );

      }

      cout << "multimap start" << endl;

      // now switch the key and value in the whole map

      //multimap<int, int, MostFreqFirst\_Compare> mymmap;

      unordered\_multimap<int, int> mymmap;

      for(auto itr = myumap.begin(); itr != myumap.end(); itr++)

          mymmap.insert(pair<int, int>(itr->second, itr->first));

      for(auto itr = mymmap.begin(); itr != mymmap.end(); itr++)

      {

          int i = itr->first;

          while (i-- >= 0)

              cout << itr->second;

      }

}

===========================================================