IT300 Assignment 2

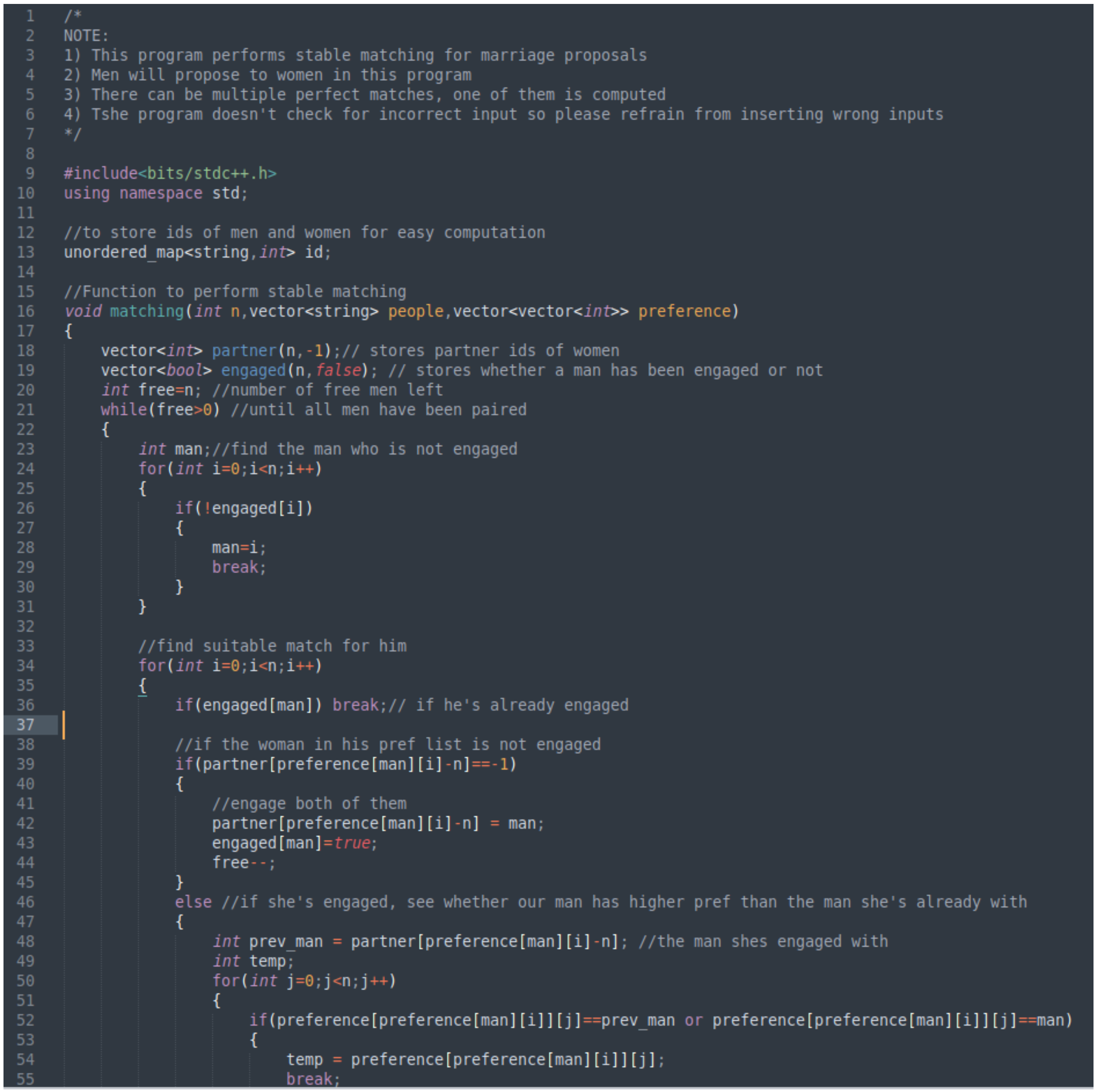
NAME: SUYASH CHINTAWAR
ROLL NO.: 191IT109
TOPIC: STABLE MATCHING AND
DIVIDE & CONQUER

NOTE: Codes have been put as multiple continued images. Please follow via line numbers if it is difficult to read.

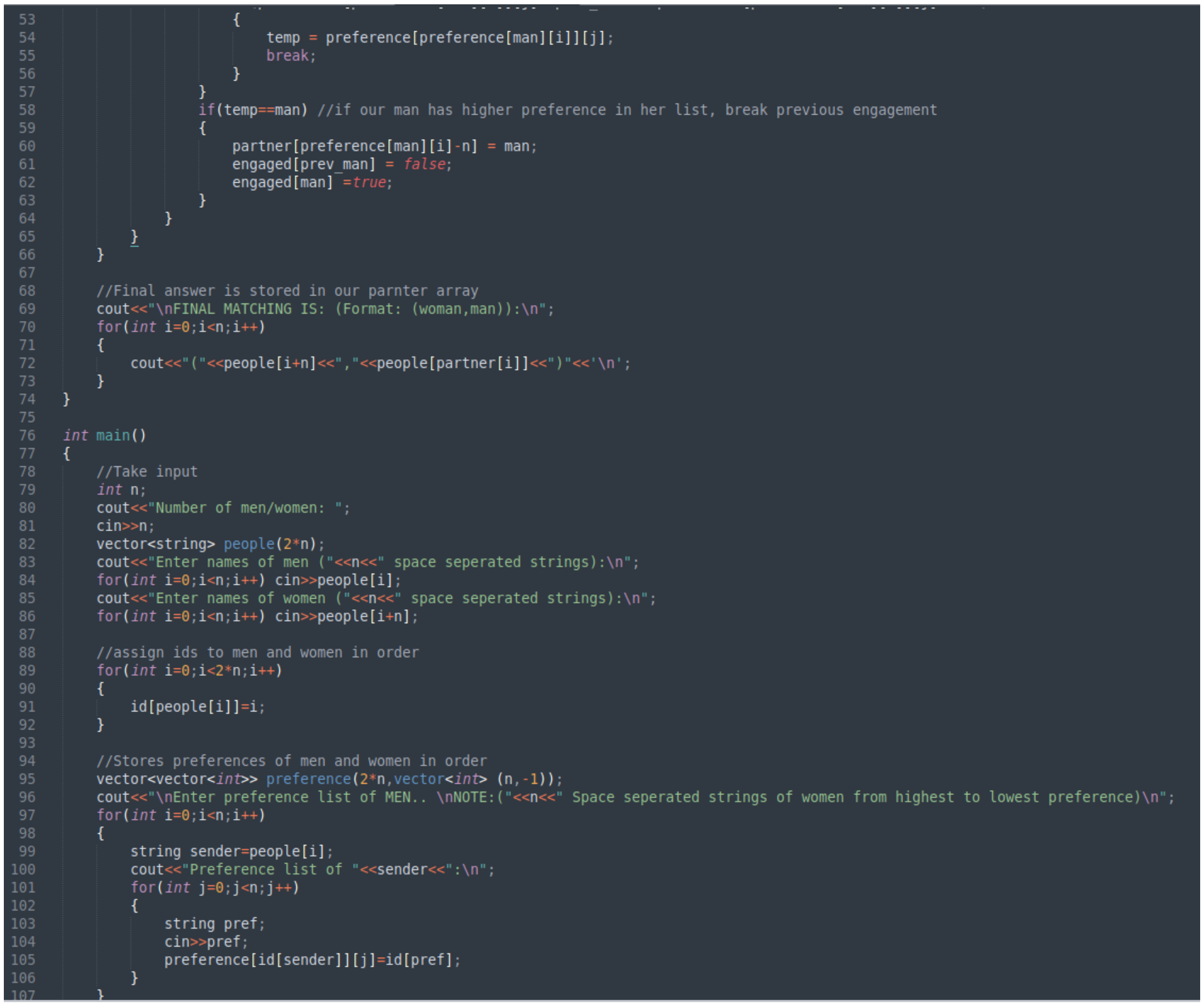
Q1. Implementation of a stable marriage problem.

SOLUTION:

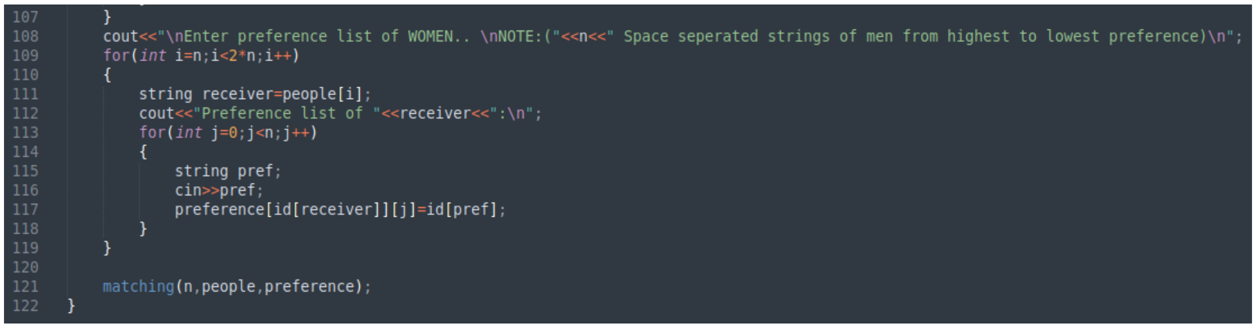
Code:



```
1   /*
2   NOTE:
3   1) This program performs stable matching for marriage proposals
4   2) Men will propose to women in this program
5   3) There can be multiple perfect matches, one of them is computed
6   4) Tshe program doesn't check for incorrect input so please refrain from inserting wrong inputs
7   */
8
9   #include<bits/stdc++.h>
10  using namespace std;
11
12  //to store ids of men and women for easy computation
13  unordered_map<string,int> id;
14
15  //Function to perform stable matching
16  void matching(int n,vector<string> people,vector<vector<int>> preference)
17  {
18      vector<int> partner(n,-1);// stores partner ids of women
19      vector<bool> engaged(n,false); // stores whether a man has been engaged or not
20      int free=n; //number of free men left
21      while(free>0) //until all men have been paired
22      {
23          int man;//find the man who is not engaged
24          for(int i=0;i<n;i++)
25          {
26              if(!engaged[i])
27              {
28                  man=i;
29                  break;
30              }
31          }
32
33          //find suitable match for him
34          for(int i=0;i<n;i++)
35          {
36              if(engaged[man]) break;// if he's already engaged
37
38              //if the woman in his pref list is not engaged
39              if(partner[preference[man][i]-n]==-1)
40              {
41                  //engage both of them
42                  partner[preference[man][i]-n] = man;
43                  engaged[man]=true;
44                  free--;
45              }
46              else //if she's engaged, see whether our man has higher pref than the man she's already with
47              {
48                  int prev_man = partner[preference[man][i]-n]; //the man shes engaged with
49                  int temp;
50                  for(int j=0;j<n;j++)
51                  {
52                      if(preference[preference[man][i]][j]==prev_man or preference[preference[man][i]][j]==man)
53                      {
54                          temp = preference[preference[man][i]][j];
55                          break;
```



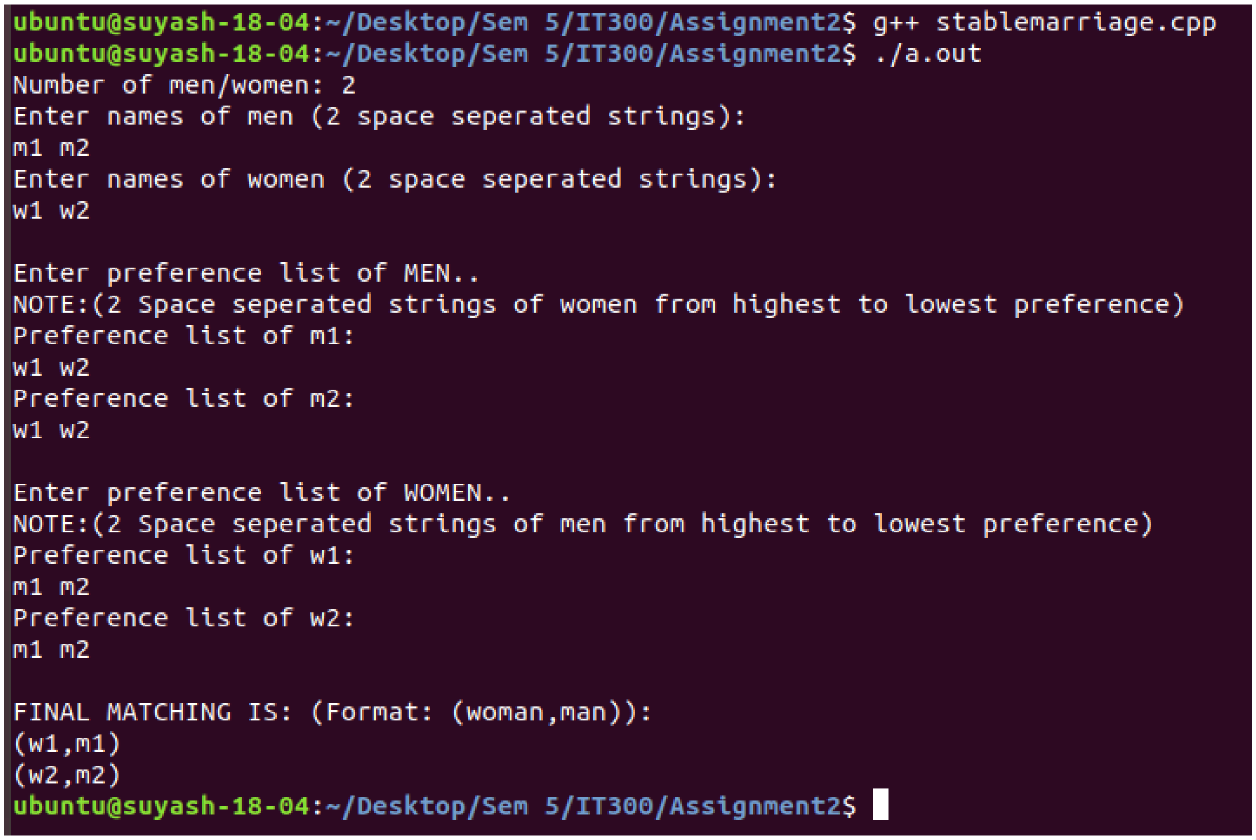
```
                    {
                        temp = preference[preference[man][i]][j];
                        break;
                    }
                }
                if(temp==man) //if our man has higher preference in her list, break previous engagement
                {
                    partner[preference[man][i]-n] = man;
                    engaged[prev_man] = false;
                    engaged[man] =true;
                }
            }
        }
    }

    //Final answer is stored in our parnter array
    cout<<"\nFINAL MATCHING IS: (Format: (woman,man)):\n";
    for(int i=0;i<n;i++)
    {
        cout<<"("<<people[i+n]<<","<<people[partner[i]]<<")"<<'\n';
    }
}

int main()
{
    //Take input
    int n;
    cout<<"Number of men/women: ";
    cin>>n;
    vector<string> people(2*n);
    cout<<"Enter names of men ("<<n<<" space seperated strings):\n";
    for(int i=0;i<n;i++) cin>>people[i];
    cout<<"Enter names of women ("<<n<<" space seperated strings):\n";
    for(int i=0;i<n;i++) cin>>people[i+n];

    //assign ids to men and women in order
    for(int i=0;i<2*n;i++)
    {
        id[people[i]]=i;
    }

    //Stores preferences of men and women in order
    vector<vector<int>> preference(2*n,vector<int> (n,-1));
    cout<<"\nEnter preference list of MEN.. \nNOTE:("<<n<<" Space seperated strings of women from highest to lowest preference)\n";
    for(int i=0;i<n;i++)
    {
        string sender=people[i];
        cout<<"Preference list of "<<sender<<":\n";
        for(int j=0;j<n;j++)
        {
            string pref;
            cin>>pref;
            preference[id[sender]][j]=id[pref];
        }
```



```
    }
    cout<<"\nEnter preference list of WOMEN.. \nNOTE:("<<n<<" Space seperated strings of men from highest to lowest preference)\n";
    for(int i=n;i<2*n;i++)
    {
        string receiver=people[i];
        cout<<"Preference list of "<<receiver<<":\n";
        for(int j=0;j<n;j++)
        {
            string pref;
            cin>>pref;
            preference[id[receiver]][j]=id[pref];
        }
    }

    matching(n,people,preference);
}
```

Fig 1. Code for Question 1 (above 3 images combined)

Output:



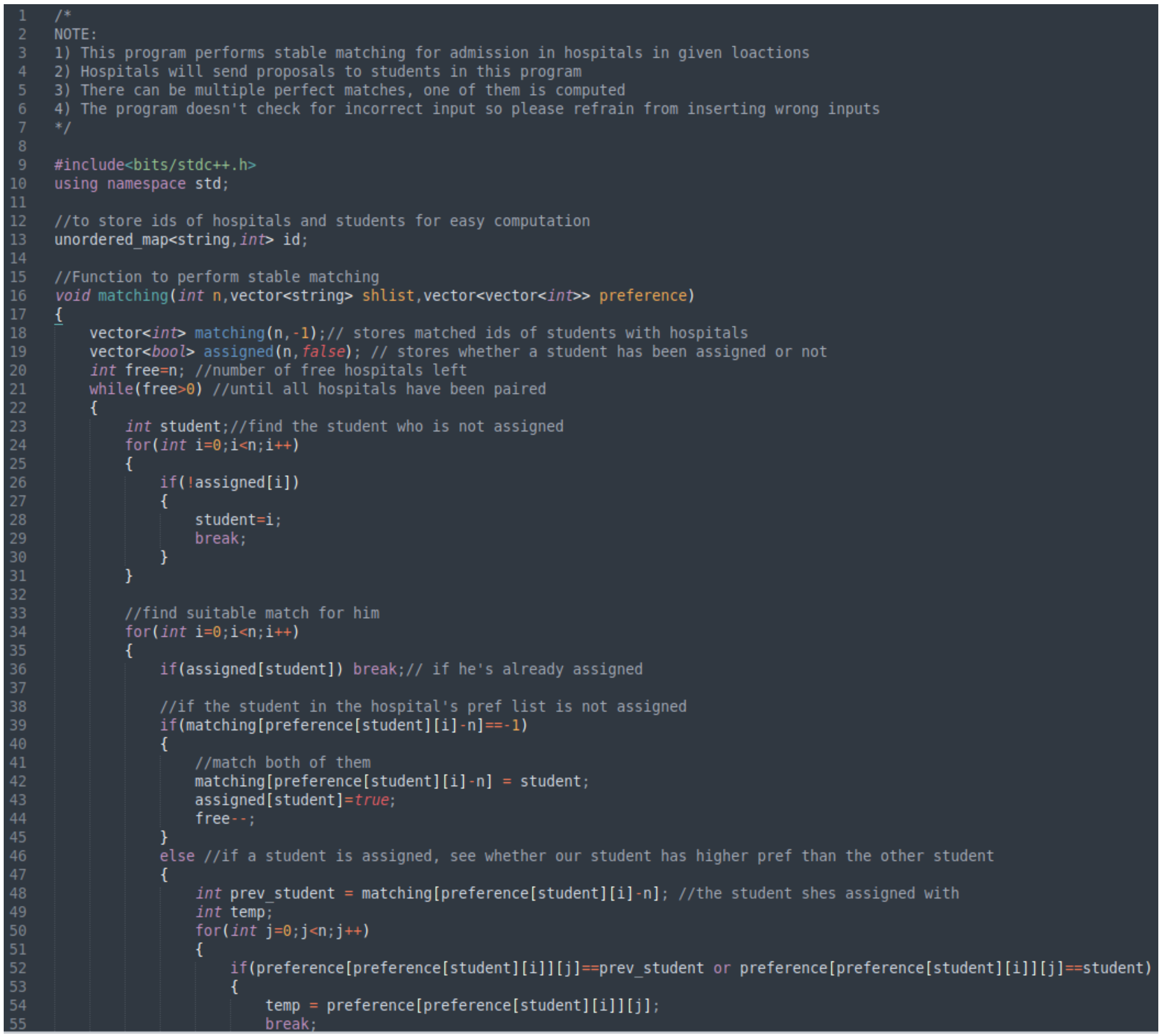
```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ g++ stablemarriage.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ ./a.out
Number of men/women: 2
Enter names of men (2 space seperated strings):
m1 m2
Enter names of women (2 space seperated strings):
w1 w2

Enter preference list of MEN..
NOTE:(2 Space seperated strings of women from highest to lowest preference)
Preference list of m1:
w1 w2
Preference list of m2:
w1 w2

Enter preference list of WOMEN..
NOTE:(2 Space seperated strings of men from highest to lowest preference)
Preference list of w1:
m1 m2
Preference list of w2:
m1 m2

FINAL MATCHING IS: (Format: (woman,man)):
(w1,m1)
(w2,m2)
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$
```

Fig 2: Output of Question 1

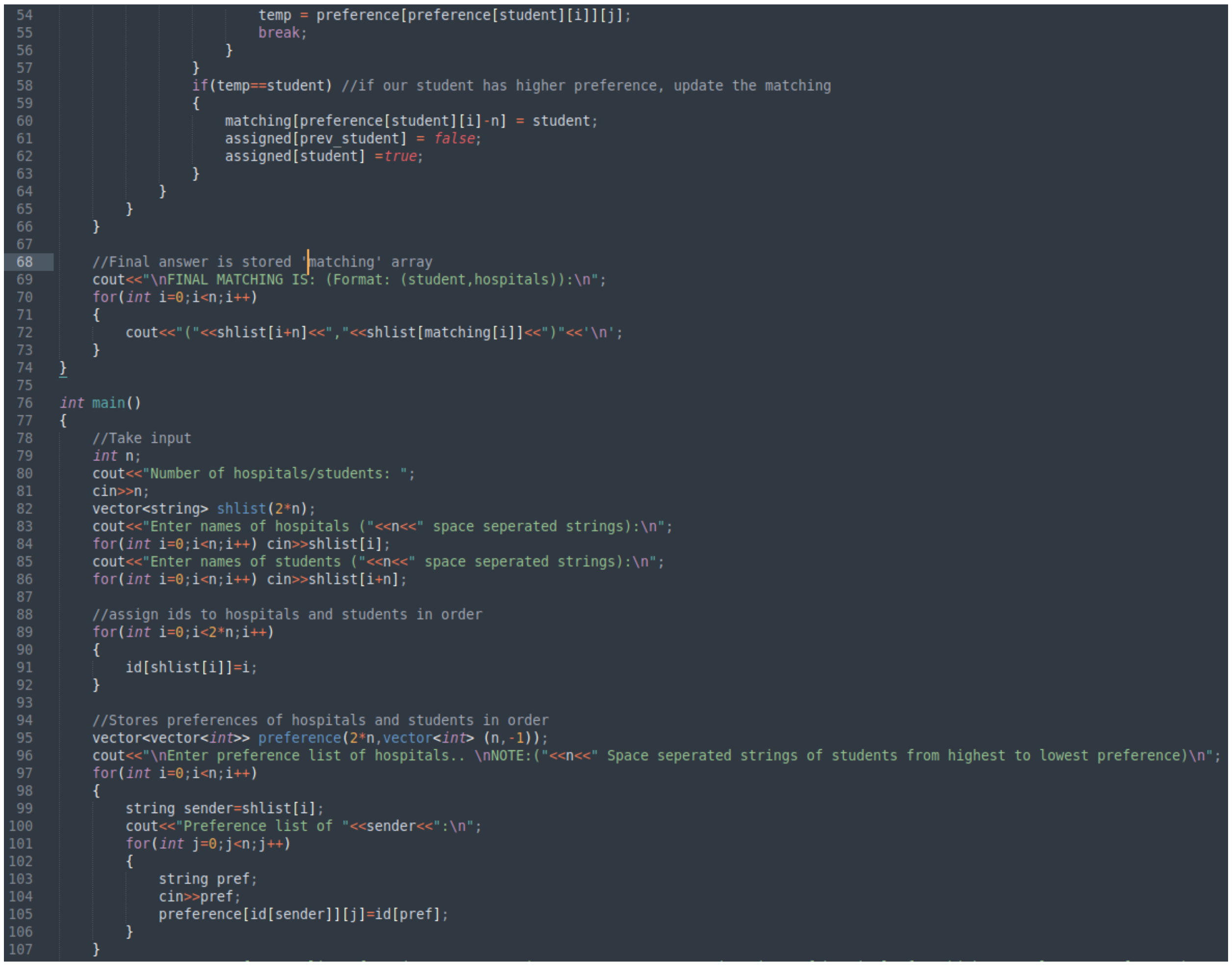
Q2. Given a set of preferences among hospitals and medical students, implement a stable matching admissions process.

SOLUTION:

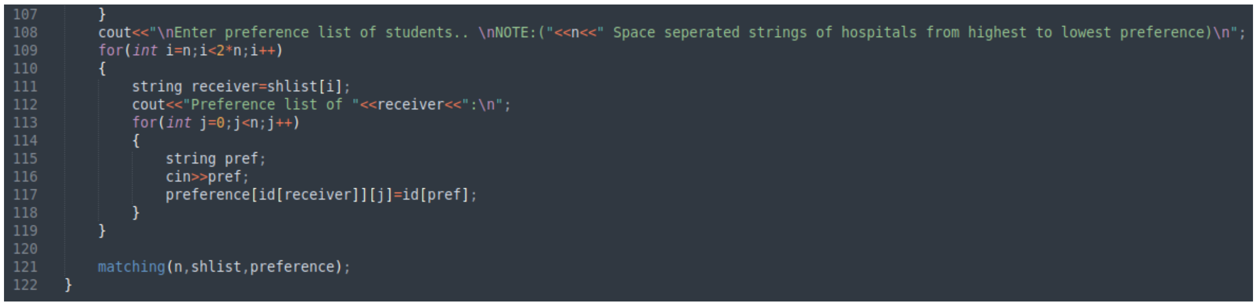
Code:



```
/*
NOTE:
1) This program performs stable matching for admission in hospitals in given loactions
2) Hospitals will send proposals to students in this program
3) There can be multiple perfect matches, one of them is computed
4) The program doesn't check for incorrect input so please refrain from inserting wrong inputs
*/

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

//to store ids of hospitals and students for easy computation
unordered_map<string,int> id;

//Function to perform stable matching
void matching(int n,vector<string> shlist,vector<vector<int>> preference)
{
    vector<int> matching(n,-1);// stores matched ids of students with hospitals
    vector<bool> assigned(n,false); // stores whether a student has been assigned or not
    int free=n; //number of free hospitals left
    while(free>0) //until all hospitals have been paired
    {
        int student;//find the student who is not assigned
        for(int i=0;i<n;i++)
        {
            if(!assigned[i])
            {
                student=i;
                break;
            }
        }

        //find suitable match for him
        for(int i=0;i<n;i++)
        {
            if(assigned[student]) break;// if he's already assigned

            //if the student in the hospital's pref list is not assigned
            if(matching[preference[student][i]-n]==-1)
            {
                //match both of them
                matching[preference[student][i]-n] = student;
                assigned[student]=true;
                free--;
            }
            else //if a student is assigned, see whether our student has higher pref than the other student
            {
                int prev_student = matching[preference[student][i]-n]; //the student shes assigned with
                int temp;
                for(int j=0;j<n;j++)
                {
                    if(preference[preference[student][i]][j]==prev_student or preference[preference[student][i]][j]==student)
                    {
                        temp = preference[preference[student][i]][j];
                        break;
```



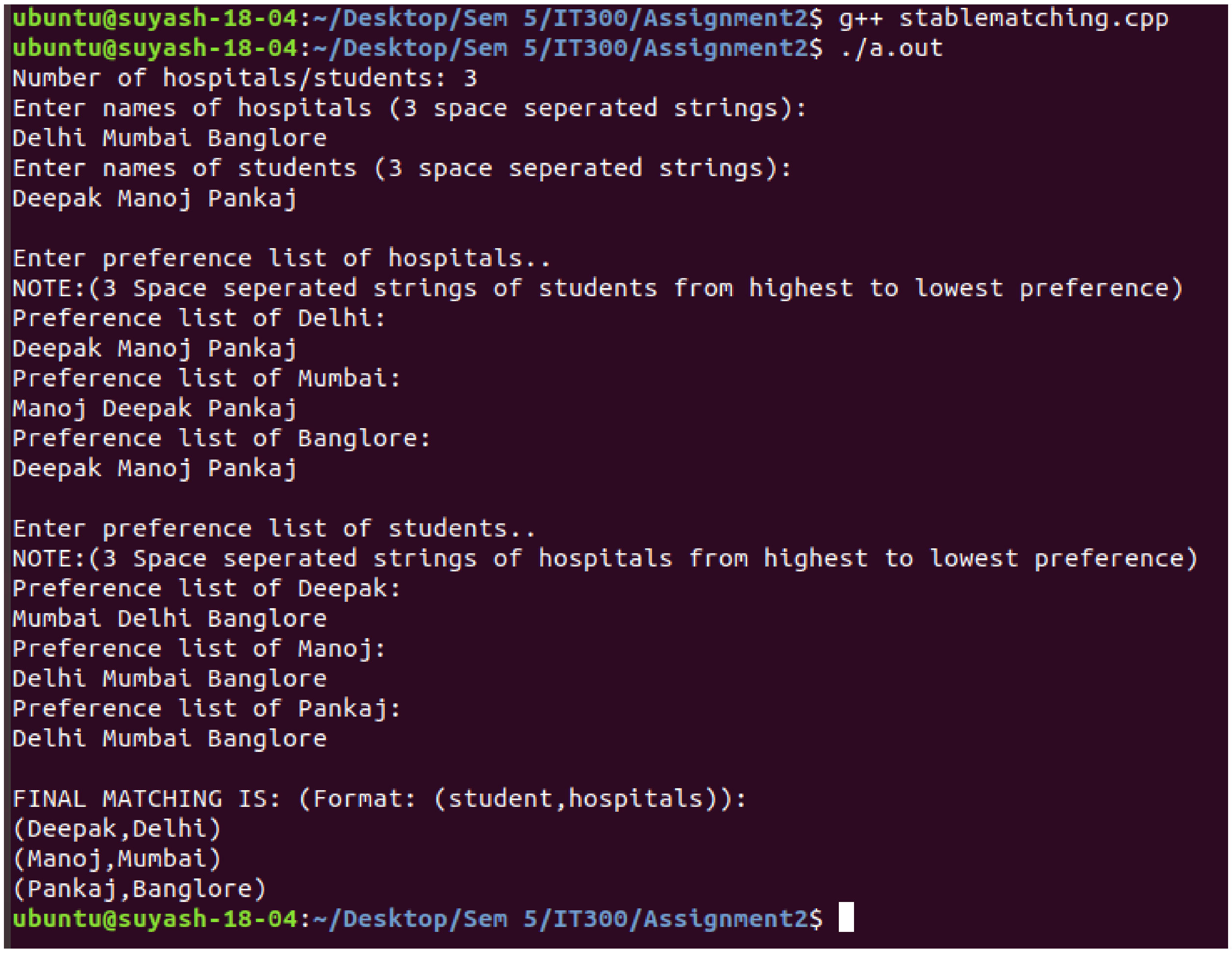
```cpp
                        temp = preference[preference[student][i]][j];
                        break;
                    }
                }
                if(temp==student) //if our student has higher preference, update the matching
                {
                    matching[preference[student][i]-n] = student;
                    assigned[prev_student] = false;
                    assigned[student] =true;
                }
            }
        }
    }

    //Final answer is stored 'matching' array
    cout<<"\nFINAL MATCHING IS: (Format: (student,hospitals)):\n";
    for(int i=0;i<n;i++)
    {
        cout<<"("<<shlist[i+n]<<","<<shlist[matching[i]]<<")"<<'\n';
    }
}

int main()
{
    //Take input
    int n;
    cout<<"Number of hospitals/students: ";
    cin>>n;
    vector<string> shlist(2*n);
    cout<<"Enter names of hospitals ("<<n<<" space seperated strings):\n";
    for(int i=0;i<n;i++) cin>>shlist[i];
    cout<<"Enter names of students ("<<n<<" space seperated strings):\n";
    for(int i=0;i<n;i++) cin>>shlist[i+n];

    //assign ids to hospitals and students in order
    for(int i=0;i<2*n;i++)
    {
        id[shlist[i]]=i;
    }

    //Stores preferences of hospitals and students in order
    vector<vector<int>> preference(2*n,vector<int> (n,-1));
    cout<<"\nEnter preference list of hospitals.. \nNOTE:("<<n<<" Space seperated strings of students from highest to lowest preference)\n";
    for(int i=0;i<n;i++)
    {
        string sender=shlist[i];
        cout<<"Preference list of "<<sender<<":\n";
        for(int j=0;j<n;j++)
        {
            string pref;
            cin>>pref;
            preference[id[sender]][j]=id[pref];
        }
    }
```



```cpp
    }
    cout<<"\nEnter preference list of students.. \nNOTE:("<<n<<" Space seperated strings of hospitals from highest to lowest preference)\n";
    for(int i=n;i<2*n;i++)
    {
        string receiver=shlist[i];
        cout<<"Preference list of "<<receiver<<":\n";
        for(int j=0;j<n;j++)
        {
            string pref;
            cin>>pref;
            preference[id[receiver]][j]=id[pref];
        }
    }

    matching(n,shlist,preference);
}
```

Fig 3. Code for Question 2 (above 3 images combined)

Output:



```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ g++ stablematching.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ ./a.out
Number of hospitals/students: 3
Enter names of hospitals (3 space seperated strings):
Delhi Mumbai Banglore
Enter names of students (3 space seperated strings):
Deepak Manoj Pankaj

Enter preference list of hospitals..
NOTE:(3 Space seperated strings of students from highest to lowest preference)
Preference list of Delhi:
Deepak Manoj Pankaj
Preference list of Mumbai:
Manoj Deepak Pankaj
Preference list of Banglore:
Deepak Manoj Pankaj

Enter preference list of students..
NOTE:(3 Space seperated strings of hospitals from highest to lowest preference)
Preference list of Deepak:
Mumbai Delhi Banglore
Preference list of Manoj:
Delhi Mumbai Banglore
Preference list of Pankaj:
Delhi Mumbai Banglore

FINAL MATCHING IS: (Format: (student,hospitals)):
(Deepak,Delhi)
(Manoj,Mumbai)
(Pankaj,Banglore)
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ 
```

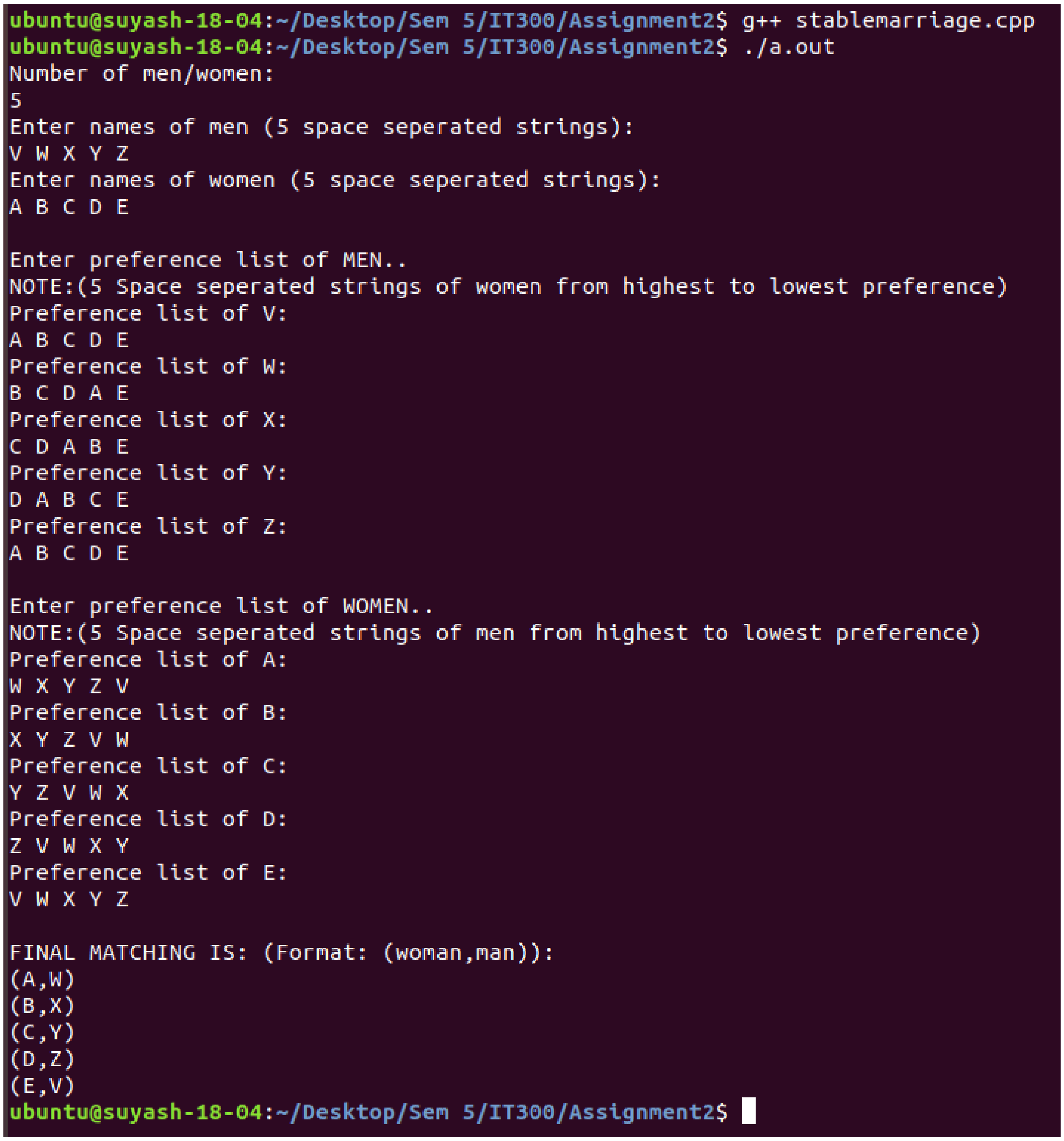
Fig 4: Output of Question 2

Q3. Implementation of propose and reject algorithm using Gale-shapley approach.

SOLUTION:

The sample output given in the problem statement is a stable matching algorithm. Hence the code will remain the same in that case.

Output:



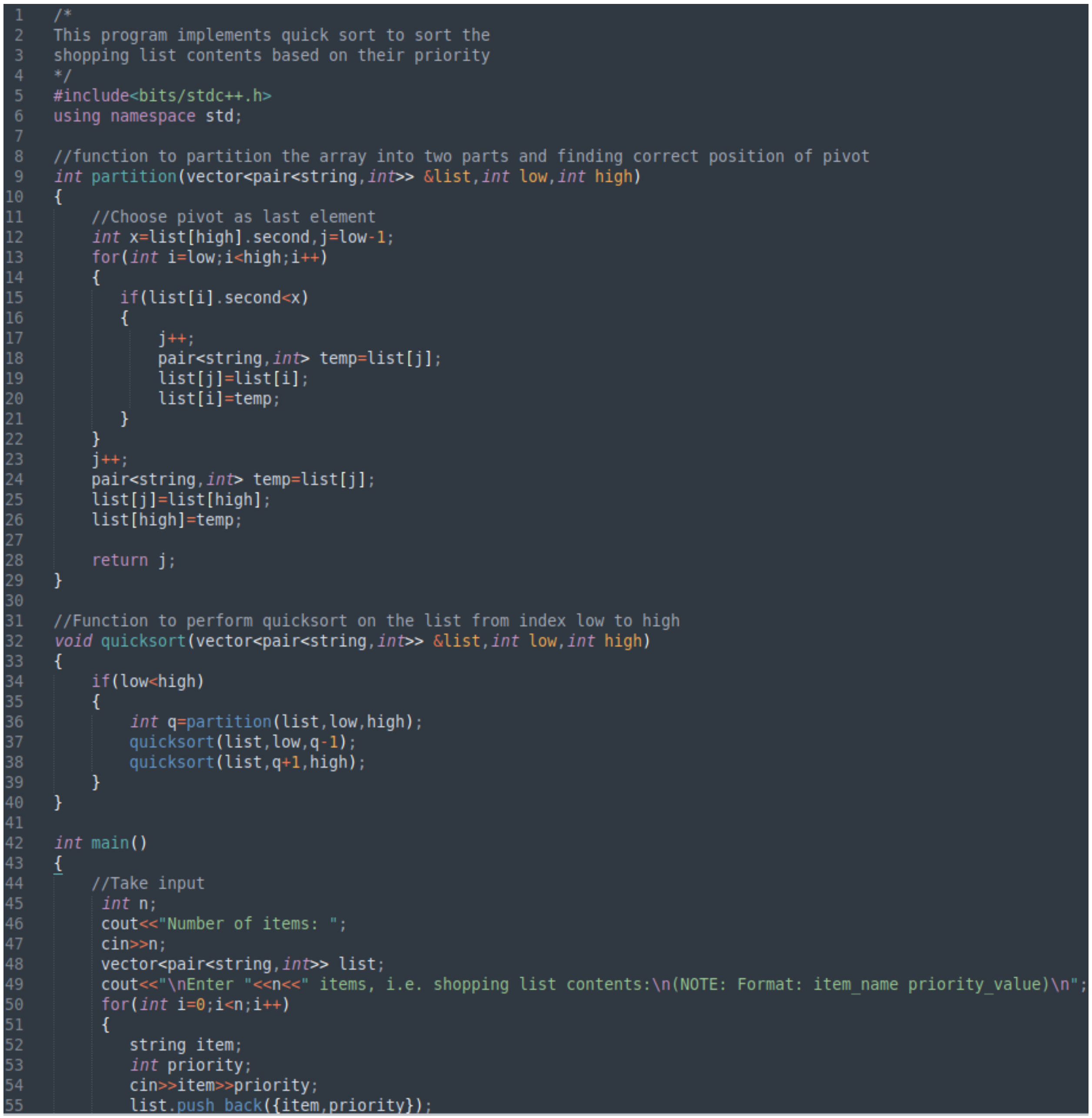
```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ g++ stablemarriage.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ ./a.out
Number of men/women:
5
Enter names of men (5 space seperated strings):
V W X Y Z
Enter names of women (5 space seperated strings):
A B C D E

Enter preference list of MEN..
NOTE:(5 Space seperated strings of women from highest to lowest preference)
Preference list of V:
A B C D E
Preference list of W:
B C D A E
Preference list of X:
C D A B E
Preference list of Y:
D A B C E
Preference list of Z:
A B C D E

Enter preference list of WOMEN..
NOTE:(5 Space seperated strings of men from highest to lowest preference)
Preference list of A:
W X Y Z V
Preference list of B:
X Y Z V W
Preference list of C:
Y Z V W X
Preference list of D:
Z V W X Y
Preference list of E:
V W X Y Z

FINAL MATCHING IS: (Format: (woman,man)):
(A,W)
(B,X)
(C,Y)
(D,Z)
(E,V)
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ 
```

Fig 5. Output for Question 3

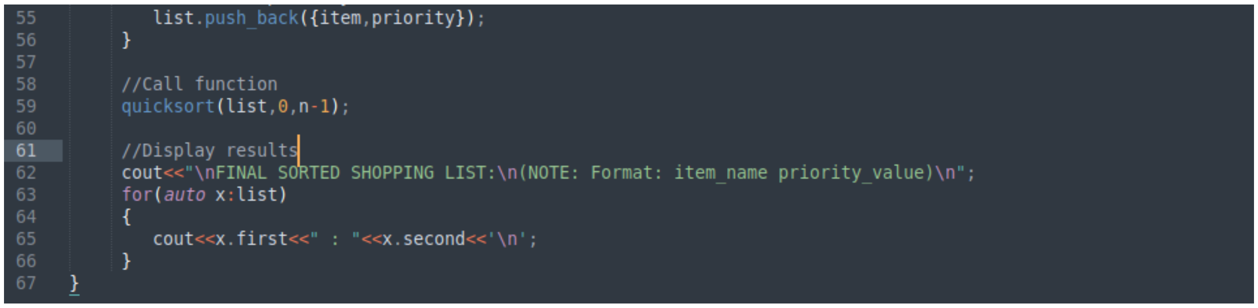
Q4. You have a shopping list, and your friend is telling you to grab them in 20 minutes. He also gives you priorities, so you need to grab them first

SOLUTION:

Code:



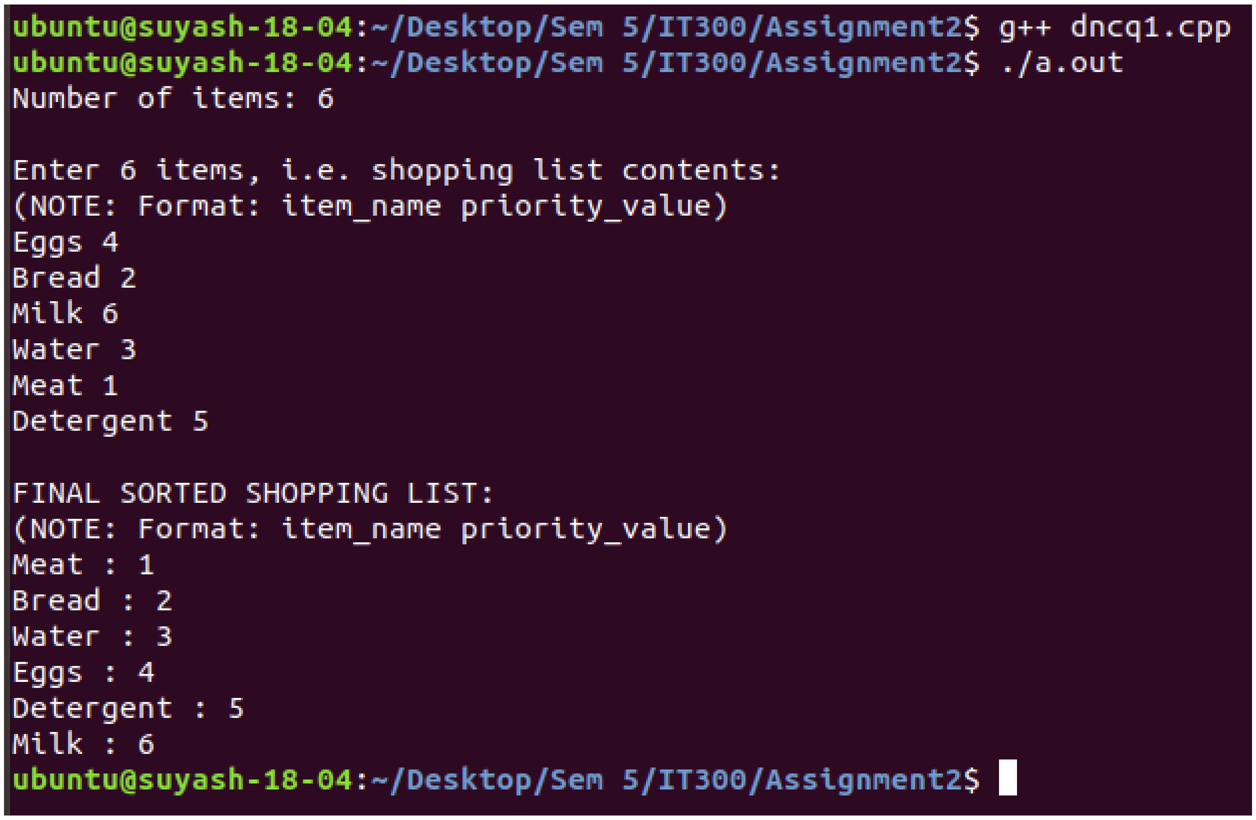
```cpp
/*
This program implements quick sort to sort the
shopping list contents based on their priority
*/
#include<bits/stdc++.h>
using namespace std;

//function to partition the array into two parts and finding correct position of pivot
int partition(vector<pair<string,int>> &list,int low,int high)
{
    //Choose pivot as last element
    int x=list[high].second,j=low-1;
    for(int i=low;i<high;i++)
    {
       if(list[i].second<x)
       {
           j++;
           pair<string,int> temp=list[j];
           list[j]=list[i];
           list[i]=temp;
       }
    }
    j++;
    pair<string,int> temp=list[j];
    list[j]=list[high];
    list[high]=temp;

    return j;
}

//Function to perform quicksort on the list from index low to high
void quicksort(vector<pair<string,int>> &list,int low,int high)
{
    if(low<high)
    {
        int q=partition(list,low,high);
        quicksort(list,low,q-1);
        quicksort(list,q+1,high);
    }
}

int main()
{
    //Take input
     int n;
     cout<<"Number of items: ";
     cin>>n;
     vector<pair<string,int>> list;
     cout<<"\nEnter "<<n<<" items, i.e. shopping list contents:\n(NOTE: Format: item_name priority_value)\n";
     for(int i=0;i<n;i++)
     {
        string item;
        int priority;
        cin>>item>>priority;
        list.push_back({item,priority});
```



```cpp
        list.push_back({item,priority});
     }

     //Call function
     quicksort(list,0,n-1);

     //Display results
     cout<<"\nFINAL SORTED SHOPPING LIST:\n(NOTE: Format: item_name priority_value)\n";
     for(auto x:list)
     {
        cout<<x.first<<" : "<<x.second<<'\n';
     }
}
```

Fig 6. Code for Question 4 (above 2 images combined)

Output:



```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ g++ dncq1.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ ./a.out
Number of items: 6

Enter 6 items, i.e. shopping list contents:
(NOTE: Format: item_name priority_value)
Eggs 4
Bread 2
Milk 6
Water 3
Meat 1
Detergent 5

FINAL SORTED SHOPPING LIST:
(NOTE: Format: item_name priority_value)
Meat : 1
Bread : 2
Water : 3
Eggs : 4
Detergent : 5
Milk : 6
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$
```

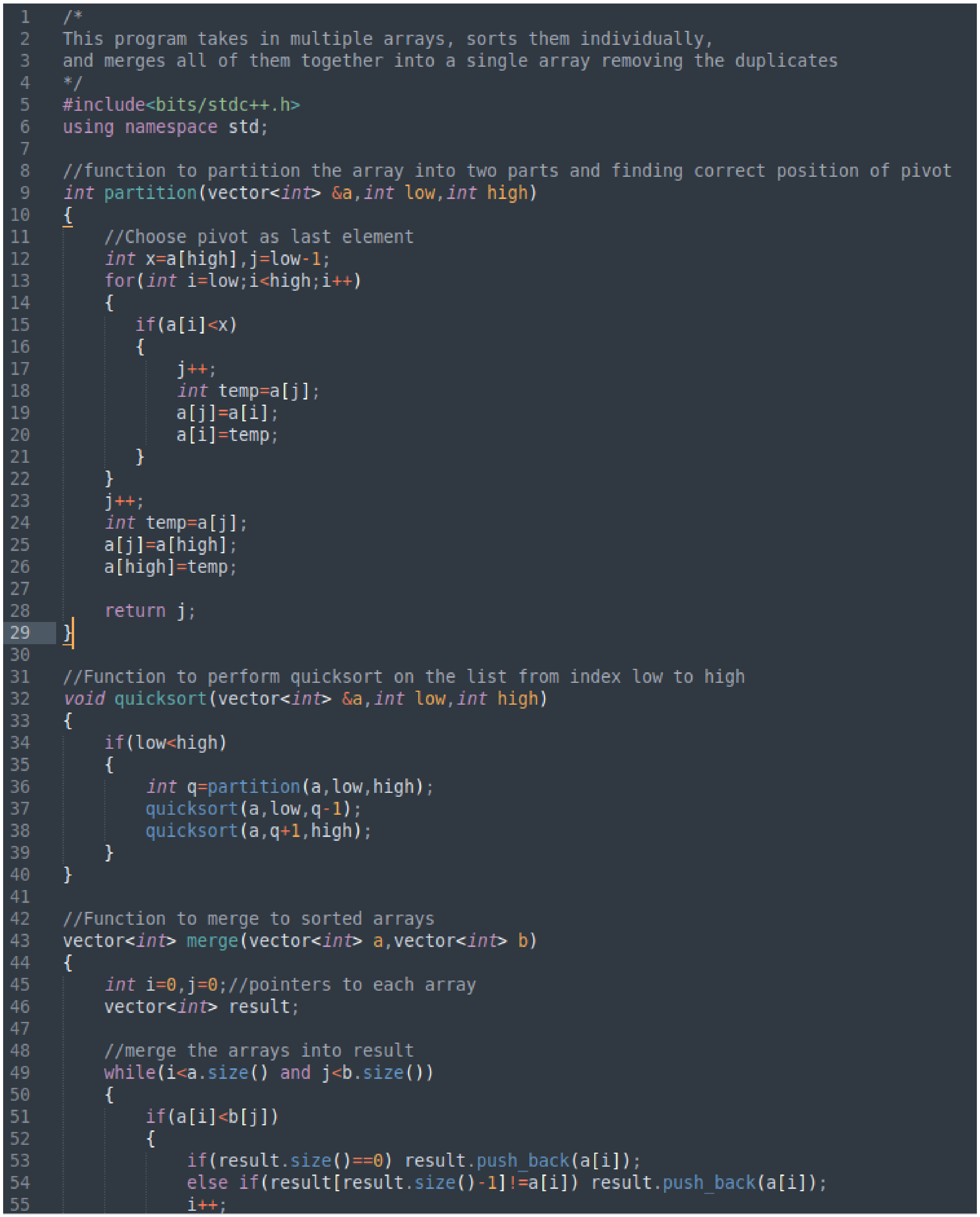
Fig 7: Output for Question 4

Q5. Suppose you had 20 years of stock market data in 300 files. you wanted to combine the files and remove duplicate data. Some of the files were not properly sorted by time. Each file was about 150MB, so you could not load all of the data into RAM at once. you only have 32GB of ram. you could use virtual memory, but it would cause lots of swapping and bring the system to its knees. How are you going to implement it?

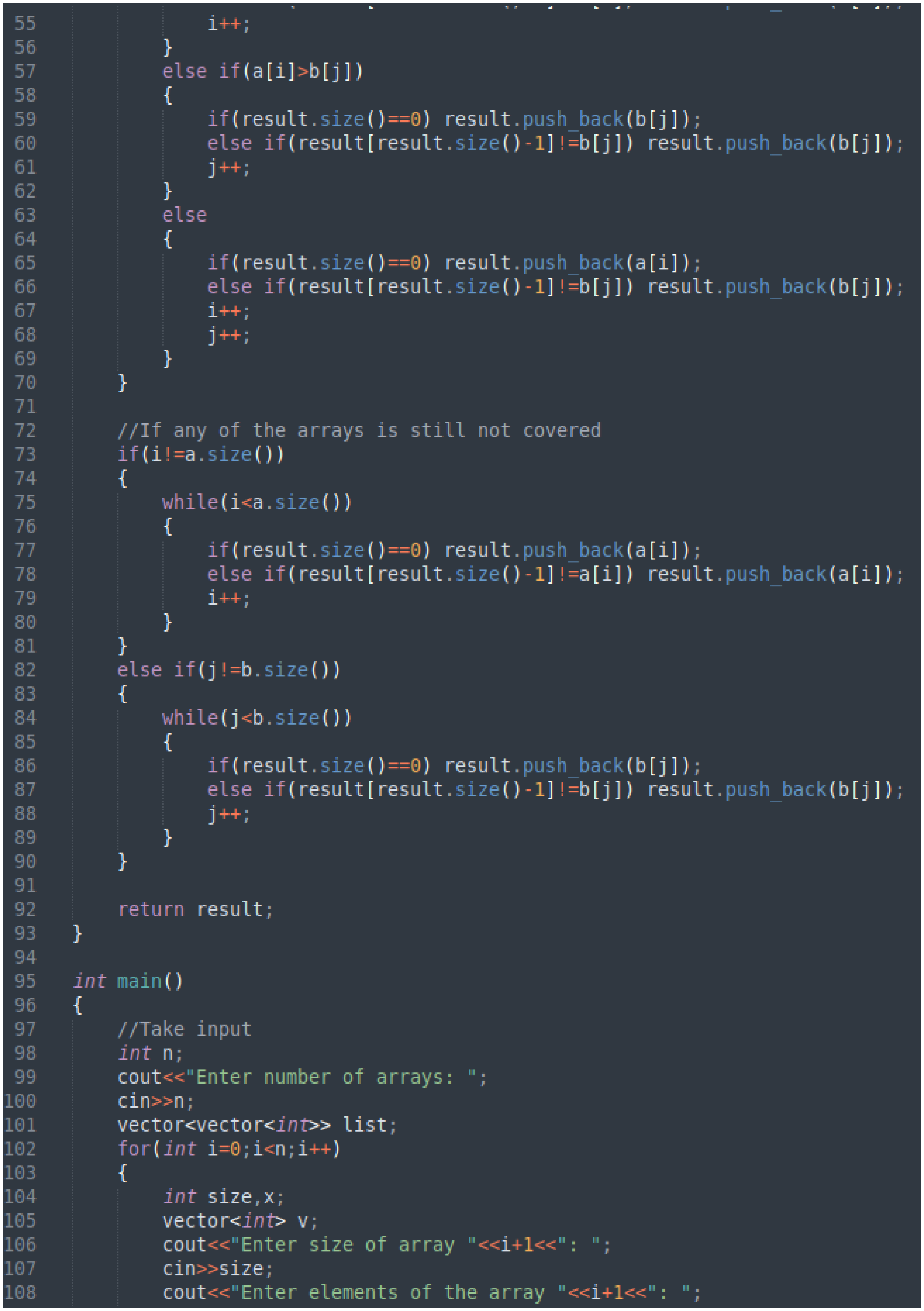
SOLUTION:

Code:

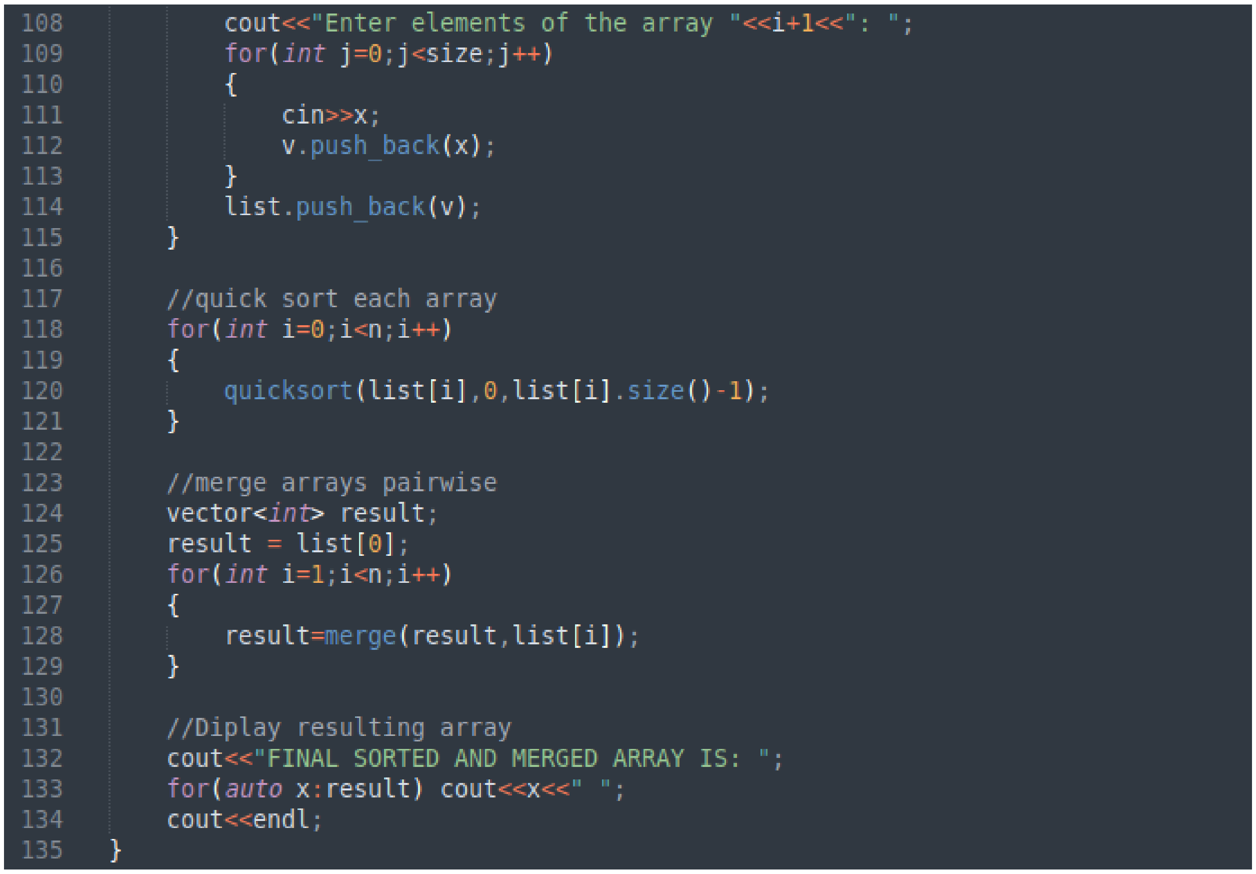
(continued...)



```cpp
/*
This program takes in multiple arrays, sorts them individually,
and merges all of them together into a single array removing the duplicates
*/
#include<bits/stdc++.h>
using namespace std;

//function to partition the array into two parts and finding correct position of pivot
int partition(vector<int> &a,int low,int high)
{
    //Choose pivot as last element
    int x=a[high],j=low-1;
    for(int i=low;i<high;i++)
    {
        if(a[i]<x)
        {
            j++;
            int temp=a[j];
            a[j]=a[i];
            a[i]=temp;
        }
    }
    j++;
    int temp=a[j];
    a[j]=a[high];
    a[high]=temp;

    return j;
}

//Function to perform quicksort on the list from index low to high
void quicksort(vector<int> &a,int low,int high)
{
    if(low<high)
    {
        int q=partition(a,low,high);
        quicksort(a,low,q-1);
        quicksort(a,q+1,high);
    }
}

//Function to merge to sorted arrays
vector<int> merge(vector<int> a,vector<int> b)
{
    int i=0,j=0;//pointers to each array
    vector<int> result;

    //merge the arrays into result
    while(i<a.size() and j<b.size())
    {
        if(a[i]<b[j])
        {
            if(result.size()==0) result.push_back(a[i]);
            else if(result[result.size()-1]!=a[i]) result.push_back(a[i]);
            i++;
```



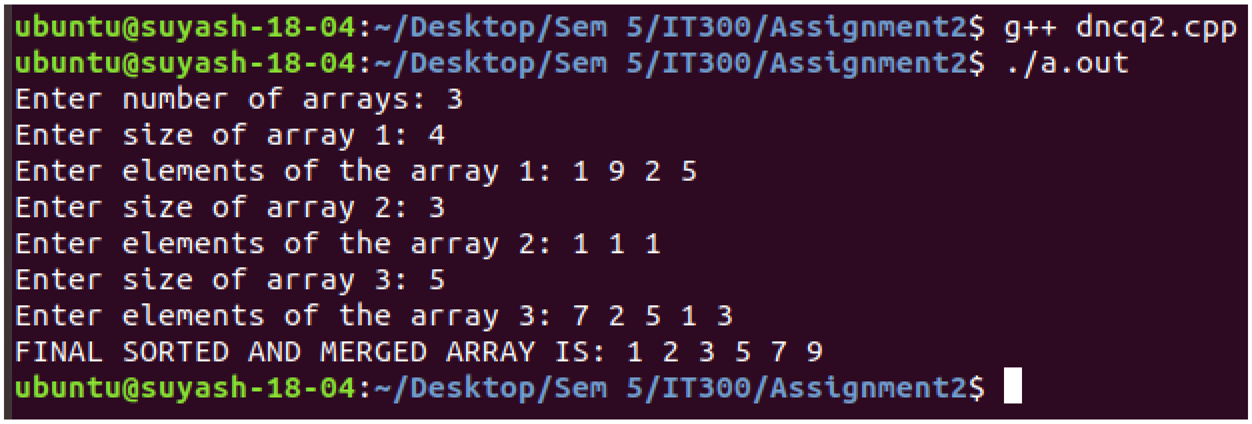
```cpp
            i++;
        }
        else if(a[i]>b[j])
        {
            if(result.size()==0) result.push_back(b[j]);
            else if(result[result.size()-1]!=b[j]) result.push_back(b[j]);
            j++;
        }
        else
        {
            if(result.size()==0) result.push_back(a[i]);
            else if(result[result.size()-1]!=b[j]) result.push_back(b[j]);
            i++;
            j++;
        }
    }

    //If any of the arrays is still not covered
    if(i!=a.size())
    {
        while(i<a.size())
        {
            if(result.size()==0) result.push_back(a[i]);
            else if(result[result.size()-1]!=a[i]) result.push_back(a[i]);
            i++;
        }
    }
    else if(j!=b.size())
    {
        while(j<b.size())
        {
            if(result.size()==0) result.push_back(b[j]);
            else if(result[result.size()-1]!=b[j]) result.push_back(b[j]);
            j++;
        }
    }

    return result;
}

int main()
{
    //Take input
    int n;
    cout<<"Enter number of arrays: ";
    cin>>n;
    vector<vector<int>> list;
    for(int i=0;i<n;i++)
    {
        int size,x;
        vector<int> v;
        cout<<"Enter size of array "<<i+1<<": ";
        cin>>size;
        cout<<"Enter elements of the array "<<i+1<<": ";
```



```cpp
            cout<<"Enter elements of the array "<<i+1<<": ";
            for(int j=0;j<size;j++)
            {
                cin>>x;
                v.push_back(x);
            }
            list.push_back(v);
        }

        //quick sort each array
        for(int i=0;i<n;i++)
        {
            quicksort(list[i],0,list[i].size()-1);
        }

        //merge arrays pairwise
        vector<int> result;
        result = list[0];
        for(int i=1;i<n;i++)
        {
            result=merge(result,list[i]);
        }

        //Diplay resulting array
        cout<<"FINAL SORTED AND MERGED ARRAY IS: ";
        for(auto x:result) cout<<x<<" ";
        cout<<endl;
    }
```

Fig 8. Code for Question 5 (above 3 images combined)

Output:

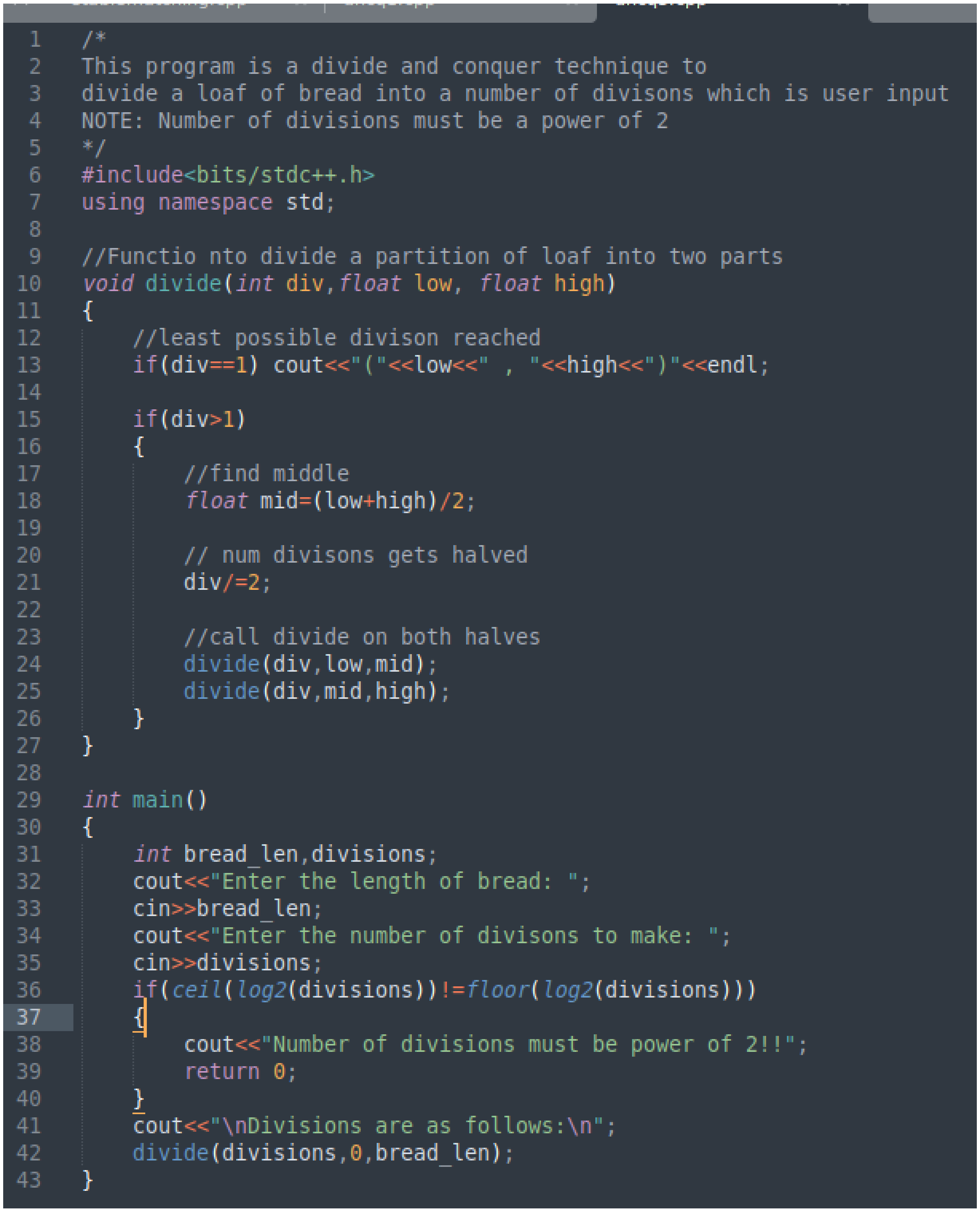


```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ g++ dncq2.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ ./a.out
Enter number of arrays: 3
Enter size of array 1: 4
Enter elements of the array 1: 1 9 2 5
Enter size of array 2: 3
Enter elements of the array 2: 1 1 1
Enter size of array 3: 5
Enter elements of the array 3: 7 2 5 1 3
FINAL SORTED AND MERGED ARRAY IS: 1 2 3 5 7 9
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$
```

Fig 9: Output of Question 5

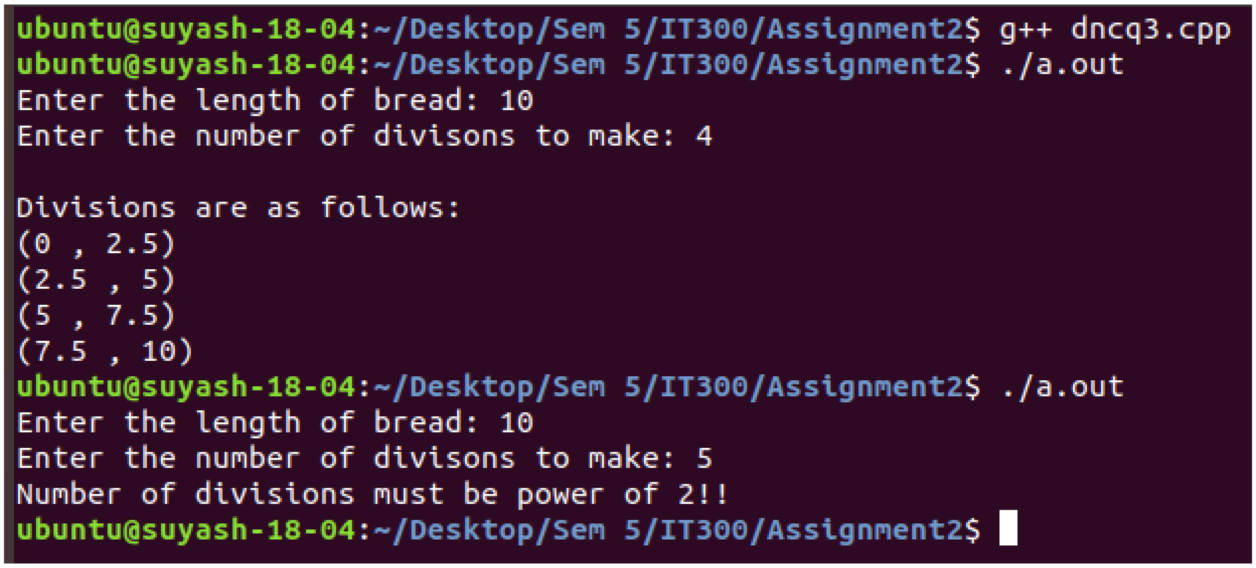
Q6. If you want to divide a long loaf of bread into 8 or 16 equal pieces, generally people cut it into two equal halves first and then cut each half into two equal halves again, repeating the process until you get as many pieces as you want -

8, 16, 32, or whatever. Almost nobody tries to divide the loaf into 8 pieces all at once - people can guess halves much better than eighths. Implement this.
SOLUTION:
Code:



```
/*
This program is a divide and conquer technique to
divide a loaf of bread into a number of divisons which is user input
NOTE: Number of divisions must be a power of 2
*/
#include<bits/stdc++.h>
using namespace std;

//Functio nto divide a partition of loaf into two parts
void divide(int div,float low, float high)
{
    //least possible divison reached
    if(div==1) cout<<"("<<low<<" , "<<high<<")"<<endl;

    if(div>1)
    {
        //find middle
        float mid=(low+high)/2;

        // num divisons gets halved
        div/=2;

        //call divide on both halves
        divide(div,low,mid);
        divide(div,mid,high);
    }
}

int main()
{
    int bread_len,divisions;
    cout<<"Enter the length of bread: ";
    cin>>bread_len;
    cout<<"Enter the number of divisons to make: ";
    cin>>divisions;
    if(ceil(log2(divisions))!=floor(log2(divisions)))
    {
        cout<<"Number of divisions must be power of 2!!";
        return 0;
    }
    cout<<"\nDivisions are as follows:\n";
    divide(divisions,0,bread_len);
}
```

Output:



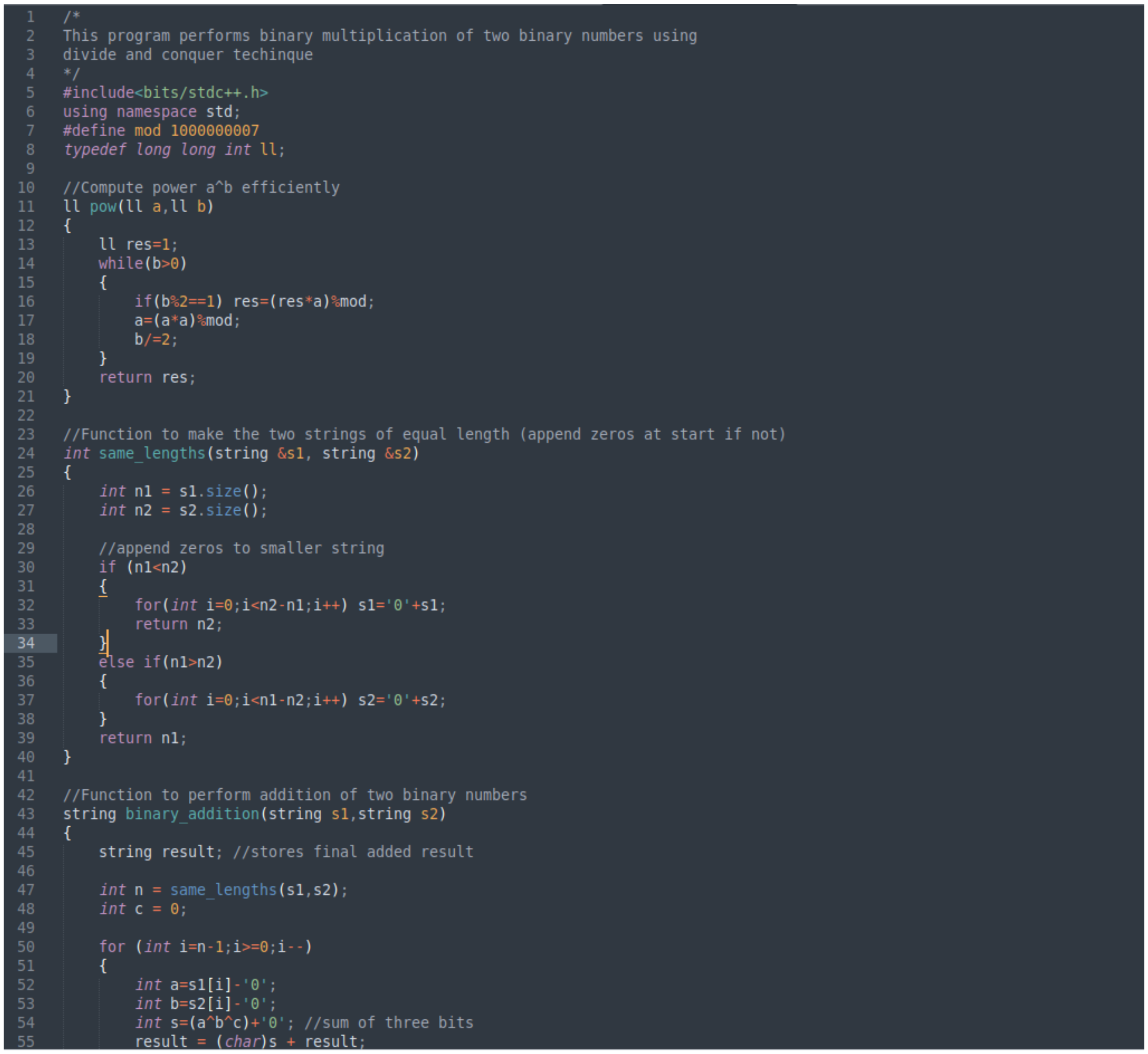
```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ g++ dncq3.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ ./a.out
Enter the length of bread: 10
Enter the number of divisons to make: 4

Divisions are as follows:
(0 , 2.5)
(2.5 , 5)
(5 , 7.5)
(7.5 , 10)
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ ./a.out
Enter the length of bread: 10
Enter the number of divisons to make: 5
Number of divisions must be power of 2!!
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ 
```

Q7. Use the divide-and-conquer integer multiplication algorithm to multiply the two binary integers 10011011 and 10111010

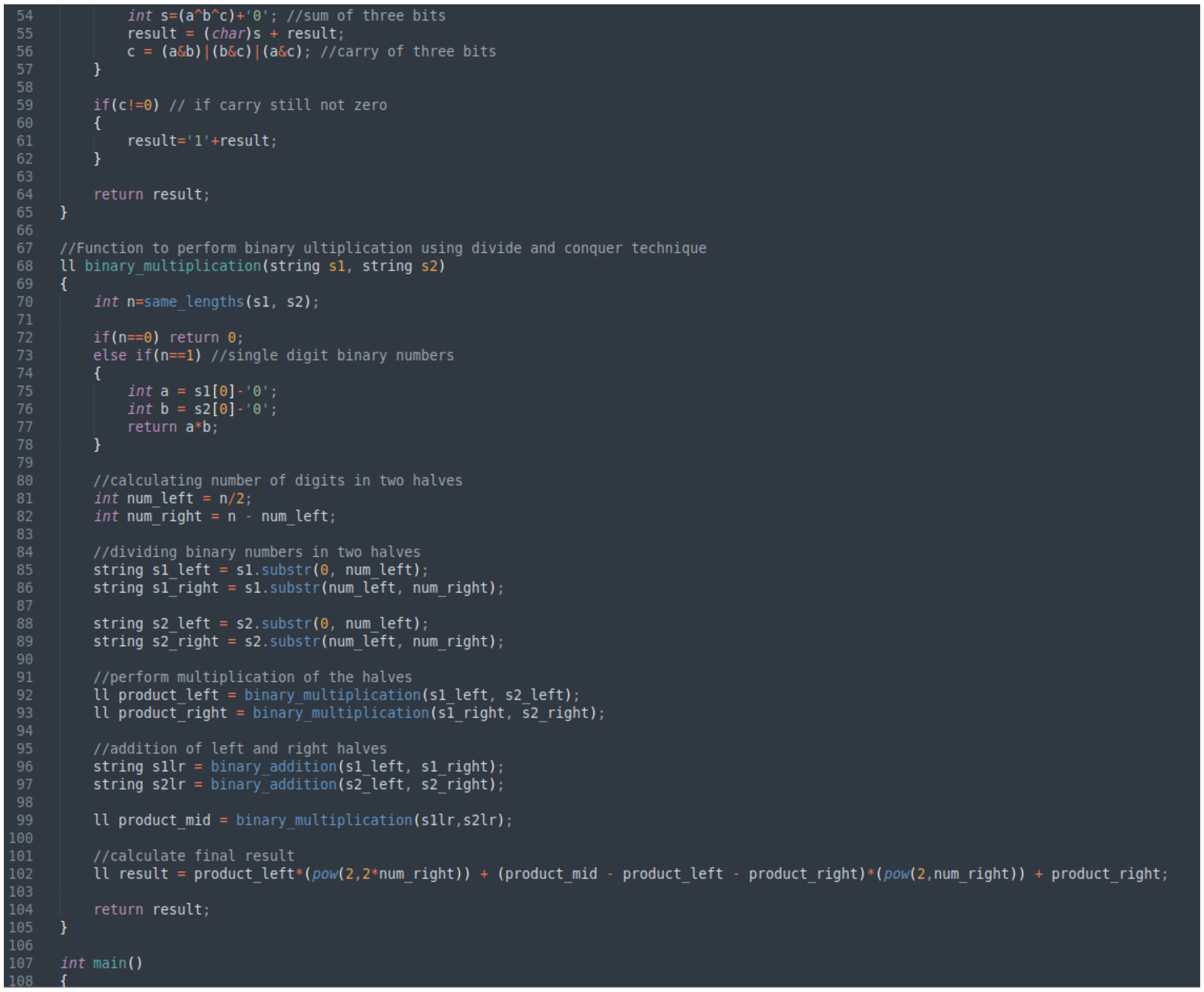
SOLUTION:

Code:

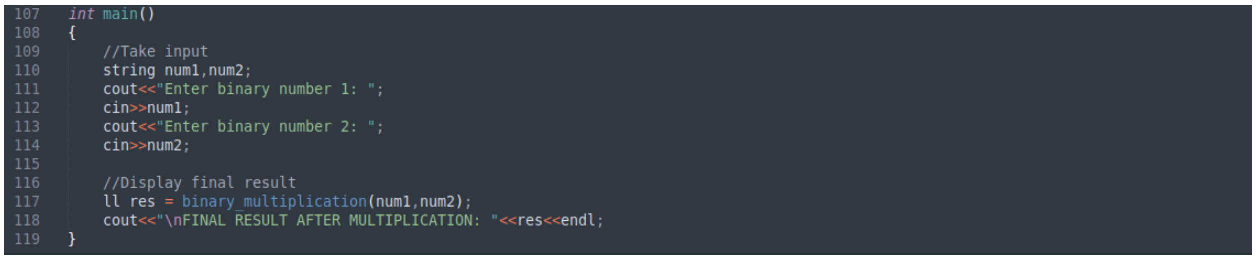
(continued...)



```cpp
/*
This program performs binary multiplication of two binary numbers using
divide and conquer techinque
*/
#include<bits/stdc++.h>
using namespace std;
#define mod 1000000007
typedef long long int ll;

//Compute power a^b efficiently
ll pow(ll a,ll b)
{
    ll res=1;
    while(b>0)
    {
        if(b%2==1) res=(res*a)%mod;
        a=(a*a)%mod;
        b/=2;
    }
    return res;
}

//Function to make the two strings of equal length (append zeros at start if not)
int same_lengths(string &s1, string &s2)
{
    int n1 = s1.size();
    int n2 = s2.size();

    //append zeros to smaller string
    if (n1<n2)
    {
        for(int i=0;i<n2-n1;i++) s1='0'+s1;
        return n2;
    }
    else if(n1>n2)
    {
        for(int i=0;i<n1-n2;i++) s2='0'+s2;
    }
    return n1;
}

//Function to perform addition of two binary numbers
string binary_addition(string s1,string s2)
{
    string result; //stores final added result

    int n = same_lengths(s1,s2);
    int c = 0;

    for (int i=n-1;i>=0;i--)
    {
        int a=s1[i]-'0';
        int b=s2[i]-'0';
        int s=(a^b^c)+'0'; //sum of three bits
        result = (char)s + result;
```

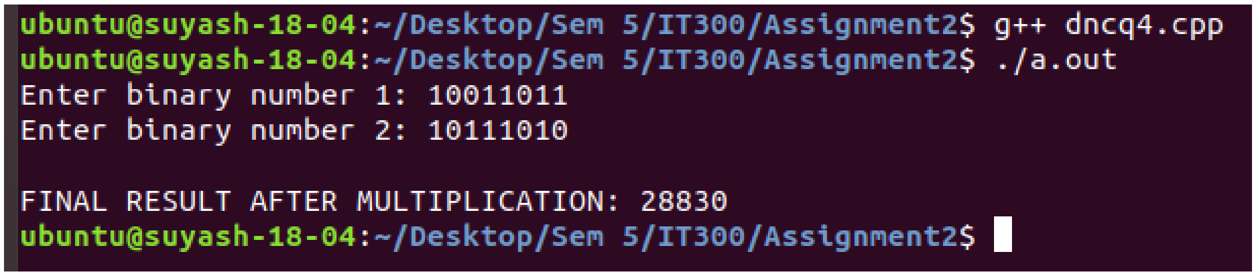


```cpp
        int s=(a^b^c)+'0'; //sum of three bits
        result = (char)s + result;
        c = (a&b)|(b&c)|(a&c); //carry of three bits
    }

    if(c!=0) // if carry still not zero
    {
        result='1'+result;
    }

    return result;
}

//Function to perform binary ultiplication using divide and conquer technique
ll binary_multiplication(string s1, string s2)
{
    int n=same_lengths(s1, s2);

    if(n==0) return 0;
    else if(n==1) //single digit binary numbers
    {
        int a = s1[0]-'0';
        int b = s2[0]-'0';
        return a*b;
    }

    //calculating number of digits in two halves
    int num_left = n/2;
    int num_right = n - num_left;

    //dividing binary numbers in two halves
    string s1_left = s1.substr(0, num_left);
    string s1_right = s1.substr(num_left, num_right);

    string s2_left = s2.substr(0, num_left);
    string s2_right = s2.substr(num_left, num_right);

    //perform multiplication of the halves
    ll product_left = binary_multiplication(s1_left, s2_left);
    ll product_right = binary_multiplication(s1_right, s2_right);

    //addition of left and right halves
    string s1lr = binary_addition(s1_left, s1_right);
    string s2lr = binary_addition(s2_left, s2_right);

    ll product_mid = binary_multiplication(s1lr,s2lr);

    //calculate final result
    ll result = product_left*(pow(2,2*num_right)) + (product_mid - product_left - product_right)*(pow(2,num_right)) + product_right;

    return result;
}

int main()
{
```



```cpp
int main()
{
    //Take input
    string num1,num2;
    cout<<"Enter binary number 1: ";
    cin>>num1;
    cout<<"Enter binary number 2: ";
    cin>>num2;

    //Display final result
    ll res = binary_multiplication(num1,num2);
    cout<<"\nFINAL RESULT AFTER MULTIPLICATION: "<<res<<endl;
}
```

Output:



```
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ g++ dncq4.cpp
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$ ./a.out
Enter binary number 1: 10011011
Enter binary number 2: 10111010

FINAL RESULT AFTER MULTIPLICATION: 28830
ubuntu@suyash-18-04:~/Desktop/Sem 5/IT300/Assignment2$
```

THANK YOU