-(2018cs94.cpp)-

```
#include <iostream>
using namespace std;
int g_c_d(int a, int b)
{
if (a == 0)
{
return b;
}
if (b == 0)
{
return a;
}
if (a == b)
{
return a;
}
if (a > b)
{
return gcd(a-b, b);
}
return gcd(a, b-a);
}
int main()
{
int a = 100, b = 50;
cout<<"GCD is "<<gcd(a, b);
return 0;
}
```

-(LCS.cpp)-

```
#include<iostream>
using namespace std;
class bigLCS{
private:
char b[200][200]{0};
int c[200][200]{0};
public:
string result=" ";
int LCS(int m,int n,string X,string Y){
if(!(m==0 || n==0))
{
for(int i=1;i<=n;i++)
{
for(int j=1;j<=m;j++)
{
if(Y[i-1]==X[j-1])
{
c[i][j]=c[i-1][j-1]+1;
b[i][j]='D';
}
else
{
c[i][j]=max(c[i-1][j],c[i][j-1]);
if(c[i-1][j]>c[i][j-1])
{
b[i][j]='U';
}
else
{
b[i][j]='L';
}
}
}
}
}
return c[n][m];
}
void print_lcs(string X,int i,int j)
{
if(!(i==0 || j==0))
{
if (b[i][j] == 'D')
{
result=X[j-1]+result;
print_lcs(X, i-1, j-1);
}
else if (b[i][j] == 'U')
{
print_lcs(X, i-1, j);
}
else
{
print_lcs(X, i, j-1);
}
}

}
};

int main()
{
Activity activities[]={{1, 4},{3, 5},{0, 6},{5, 7},{3, 8},{5, 9},{6, 10},{8, 11},{8, 12},{2, 13},{12, 14}};
Activity selected_activities[sizeof(activities)/sizeof(activities[0])]{};
int n = sizeof(activities)/sizeof(activities[0]);
int t=1;
bigLCS l;
string p,q;
int m,n,len;
cout<<"\tEnter 1st String in UpperCase : ";
cin>>p;
cout<<"\tEnter 2nd String in UpperCase : ";
cin>>q;
m=p.length();
n=q.length();
len=l.LCS(m,n,p,q);
l.print_lcs(p,n,m);
cout<<"\nTotal Length of Subsequence = "<<len;
cout<<"\nOptimized String from LCS = "<<l.result;
return 0;
}
```