Name of the Assignment: Coding line indentation program using OOP concept and software design.

Code:

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

class helper
{
private:
    int _lengthPerLine;
public:
    /** Set the length per line **/
         helper(int len)
         {
             _lengthPerLine=len;
         }
         /** Return length per line **/
         int getLen()
         {
             return _lengthPerLine;
         }
         string s;
         /** Take input from user **/
         void inp()
         {
             getline(cin,s);
         }
         /**Stores the resultant string for each line **/
         vector<string>res;
         /** Stores the starting position of each line **/
         vector<int>pos;
         /** process the input for required indentetion **/
         void process()
         {
             int flag=0;
             /** Strores the string for required length **/
             string tmp;
             /**stores each word **/
             string word;
             /**Stores the starting position of each line **/
             int start=-1;
             for(int i=0; i<s.size(); i++)
```

```
            {
                if(s[i]==' ' and !flag)
                {
                    continue;
                }
                ++flag;

                if(start == -1) start=i+1;
                if(s[i] == ' ')
                {
                    if(tmp.size()) tmp+=' ';
                    if(tmp.size()+word.size()>getLen())
                    {
                        res.push_back(tmp);
                        pos.push_back(start);
                        tmp.clear();
                        tmp+=word;
                        start=i+1-word.size();
                        word.clear();
                    }
                    else if(tmp.size()+word.size()==getLen())
                    {
                        tmp+=word;
                        res.push_back(tmp);
                        pos.push_back(start);
                        word.clear();
                        tmp.clear();
                        start=i+2;
                    }
                    else
                    {
                        tmp+=word;
                        word.clear();
                    }
                    continue;
                }
                word+=s[i];
                if(i==s.size()-1)
                {
                    pos.push_back(start);
                    if(tmp.size()) tmp+=' ';
                    if(tmp.size()+word.size()>getLen())
                    {
                        res.push_back(tmp);
                        start+=tmp.size();
```

```
                        pos.push_back(start);
                        res.push_back(word);
                    }
                    else
                    {
                        tmp+=word;
                        res.push_back(tmp);
                    }
                }
            }

        }
        /** Print the result **/
        void print()
        {
            for(int i=0; i<res.size(); i++)
            {

                string ss=res[i];
                while(ss.back()==' ') ss.pop_back();
                cout<<pos[i]<<" "<<pos[i]+ss.size()-1<<endl;
                reverse(ss.begin(),ss.end());
                while(ss.size()!=getLen()) ss+=' ';
                reverse(ss.begin(),ss.end());
                cout<<ss<<endl;
            }
        }

    };

    int main()
    {
        helper work(10);
        work.inp();
        work.process();
        work.print();
    }
```

Input & Output :