

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

Code:

```
1. #include<bits/stdc++.h>
2. using namespace std;
3.
4. class helper
5. {
6. private:
7.     int _lengthPerLine;
8. public:
9.     /** Set the length per line */
10.    helper(int len)
11.    {
12.        _lengthPerLine=len;
13.    }
14.    /** Return length per line */
15.    int getLen()
16.    {
17.        return _lengthPerLine;
18.    }
19.    string s;
20.    /** Take input from user */
21.    void inp()
22.    {
23.        getline(cin,s);
24.    }
25.    /**Stores the resultant string for each line */
26.    vector<string>res;
27.    /** Stores the starting position of each line */
28.    vector<int>pos;
29.    /** process the input for required indentetion */
30.    void process()
31.    {
32.        int flag=0;
33.        /** Stores the string for required length */
34.        string tmp;
35.        /**stores each word */
36.        string word;
37.        /**Stores the starting position of each line */
38.        int start=-1;
39.        for(int i=0; i<s.size(); i++)
```

```

40.         {
41.             if(s[i]==' ' and !flag)
42.             {
43.                 continue;
44.             }
45.             ++flag;
46.
47.             if(start == -1) start=i+1;
48.             if(s[i] == ' ')
49.             {
50.                 if(tmp.size()) tmp+=' ';
51.                 if(tmp.size()+word.size()>getLen())
52.                 {
53.                     res.push_back(tmp);
54.                     pos.push_back(start);
55.                     tmp.clear();
56.                     tmp+=word;
57.                     start=i+1-word.size();
58.                     word.clear();
59.                 }
60.                 else if(tmp.size()+word.size()==getLen())
61.                 {
62.                     tmp+=word;
63.                     res.push_back(tmp);
64.                     pos.push_back(start);
65.                     word.clear();
66.                     tmp.clear();
67.                     start=i+2;
68.                 }
69.                 else
70.                 {
71.                     tmp+=word;
72.                     word.clear();
73.                 }
74.                 continue;
75.             }
76.             word+=s[i];
77.             if(i==s.size()-1)
78.             {
79.                 pos.push_back(start);
80.                 if(tmp.size()) tmp+=' ';
81.                 if(tmp.size()+word.size()>getLen())
82.                 {
83.                     res.push_back(tmp);
84.                     start+=tmp.size();

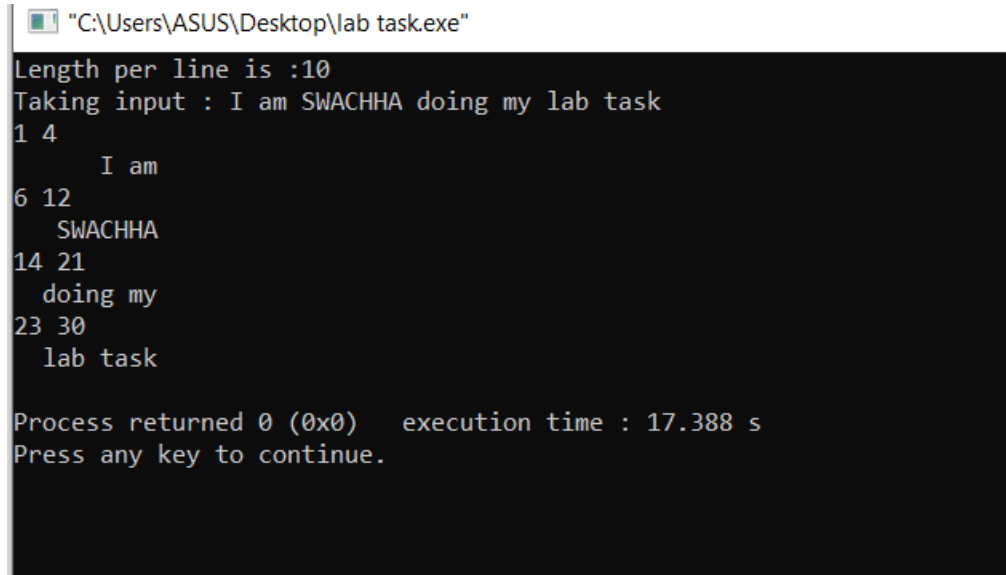
```

```

85.             pos.push_back(start);
86.             res.push_back(word);
87.         }
88.         else
89.         {
90.             tmp+=word;
91.             res.push_back(tmp);
92.         }
93.     }
94. }
95.
96. }
97. /** Print the result **/
98. void print()
99. {
100.     for(int i=0; i<res.size(); i++)
101.     {
102.
103.         string ss=res[i];
104.         while(ss.back()==' ') ss.pop_back();
105.         cout<<pos[i]<<" "<<pos[i]+ss.size()-1<<endl;
106.         reverse(ss.begin(),ss.end());
107.         while(ss.size()!=getLen()) ss+=' ';
108.         reverse(ss.begin(),ss.end());
109.         cout<<ss<<endl;
110.     }
111. }
112.
113. };
114.
115. int main()
116. {
117.     helper work(10);
118.     work.inp();
119.     work.process();
120.     work.print();
121. }

```

Input & Output :



```
"C:\Users\ASUS\Desktop\lab task.exe"
Length per line is :10
Taking input : I am SWACHHA doing my lab task
1 4
    I am
6 12
    SWACHHA
14 21
    doing my
23 30
    lab task

Process returned 0 (0x0)   execution time : 17.388 s
Press any key to continue.
```