Formatting Text Files

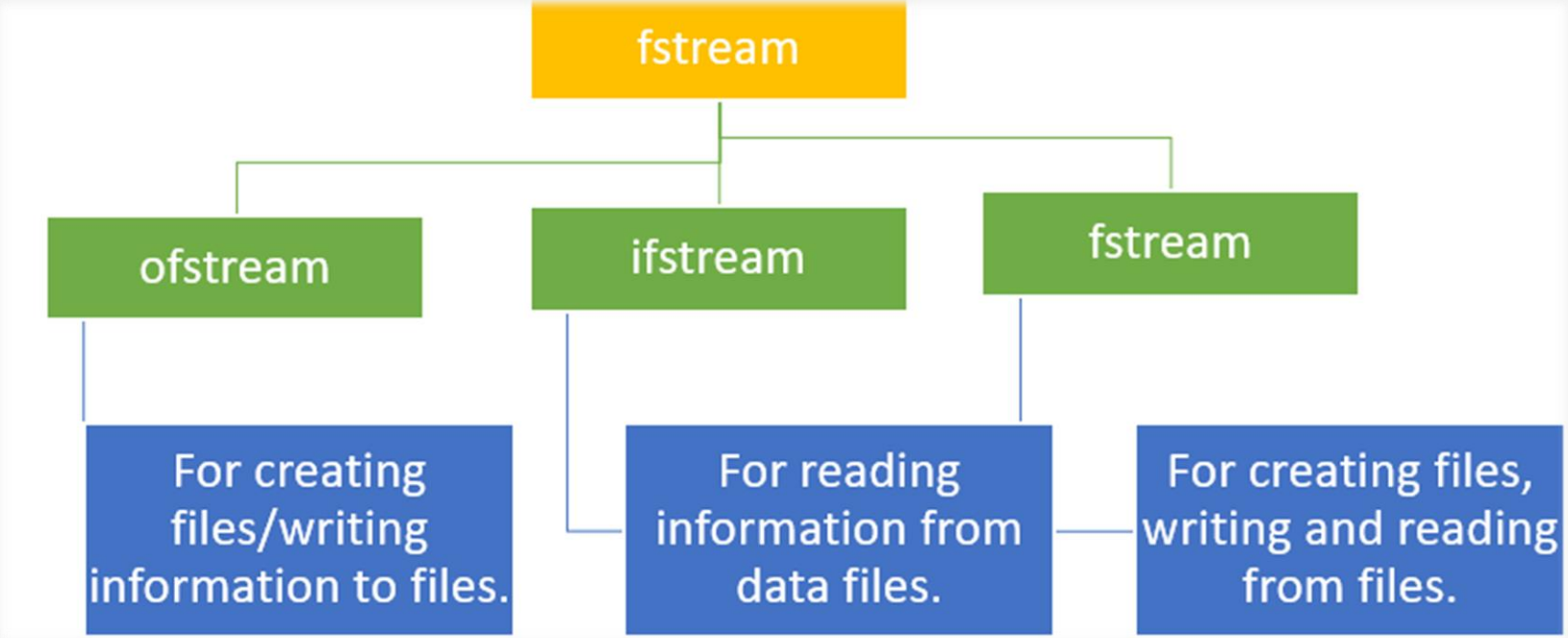
PASS I ASSEMBLER

Course2 Week-2

FSTREAM LIBRARY

The fstream library provides C++ programmers with three classes for working with files. These classes include:

- ofstream– This class represents an output stream. It's used for creating files and writing information to files.
- ifstream– This class represents an input stream. It's used for reading information from data files.
- fstream– This class generally represents a file stream. It comes with ofstream/ifstream capabilities. This means it's capable of creating files, writing to files, reading from data files.



OPEN FILES

How to Open Files:

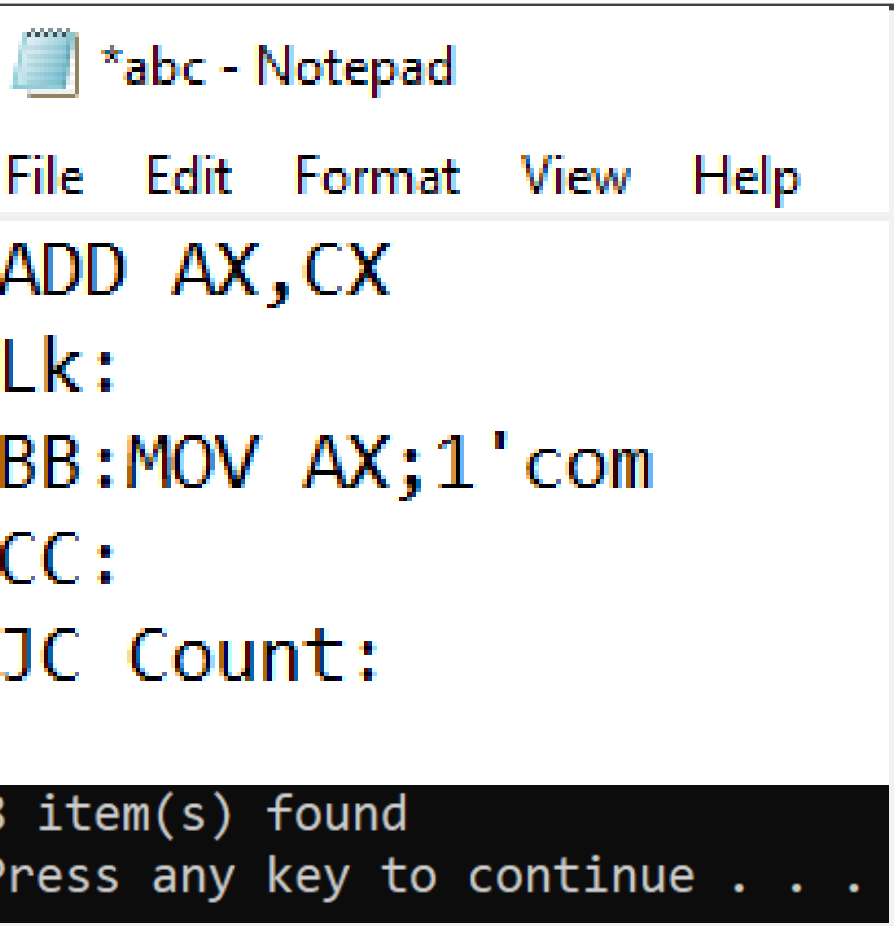
Syntax:

open (file_name, mode);

Value	Description
ios:: app	The Append mode. The output sent to the file is appended to it.
ios::ate	It opens the file for the output then moves the read and write control to file's end.
ios::in	It opens the file for a read.
ios::out	It opens the file for a write.
ios::trunc	If a file exists, the file elements should be truncated prior to its opening.

FSTREAM EXAMPLES

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;
int main () {

    ifstream ifile;
    ifile.open("abc.txt");

    if (ifile.fail()){
    cout<< "error in file"<< endl;
    exit(1);
    }
        string no_of_lines;
    int i=0;
    while (!ifile.eof()){
        ifile >> no_of_lines;
        i++;
    }
    cout << i <<" " << "item(s) found" << endl;
system("pause");
return 0;
}
```

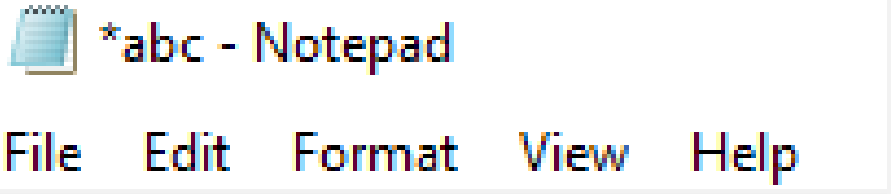


We got a problem here?

Do you have any suggestions for resolving the considered spacing issue?

FSTREAM EXAMPLES "IFSTREAM"

```cpp
#include <iostream>
#include <fstream>
#include <string>

using namespace std;
int main () {

    ifstream ifile;
    ifile.open("abc.txt");
        if (ifile.fail()){
    cout<< "error in file"<< endl;
    exit(1);
    }
        string no_of_lines;
    int i=0;
    while (!ifile.eof()){
        getline(ifile, no_of_lines);
        //ifile >> no_of_lines;
        i++;
    }
    cout << i <<" " << "item(s) found" << endl;

system("pause");
return 0;
}
```




```
5 item(s) found
Press any key to continue . . .
```

A problem can be solved by reading the whole line using:

getline(nameFileout, item_name)

FSTREAM EXAMPLES

```cpp
#include <iostream>
#include <fstream>
#include <string>

using namespace std;
int main () {
//cout << setw(10);
    ifstream ifile;
    ifile.open("abc.txt");
        if (ifile.fail()){
    cout<< "error in file"<< endl;
    exit(1);
    }
        string no_of_lines;
    int i=0;
    while (!ifile.eof()){
        getline(ifile, no_of_lines);
        //ifile >> no_of_lines;
        cout << no_of_lines << endl;
    }
    system("pause");
    return 0;
    }
```

```
ADD AX,CX
Lk:
BB:MOV AX;1'com
CC:
JC Count:
Press any key to continue
```

```cpp
#include <fstream>
#include <string>
//#include <iomanip>
using namespace std;
int main () {
//cout << setw(10);
    ifstream ifile;
    ifile.open("abc.txt");

    if (ifile.fail()){
    cout<< "error in file"<< endl;
    exit(1);
    }
        string no_of_lines;
    int i=0;
    while (!ifile.eof()){
        ifile >> no_of_lines;
        cout << no_of_lines << endl;
    }

system("pause");
return 0;
}
```

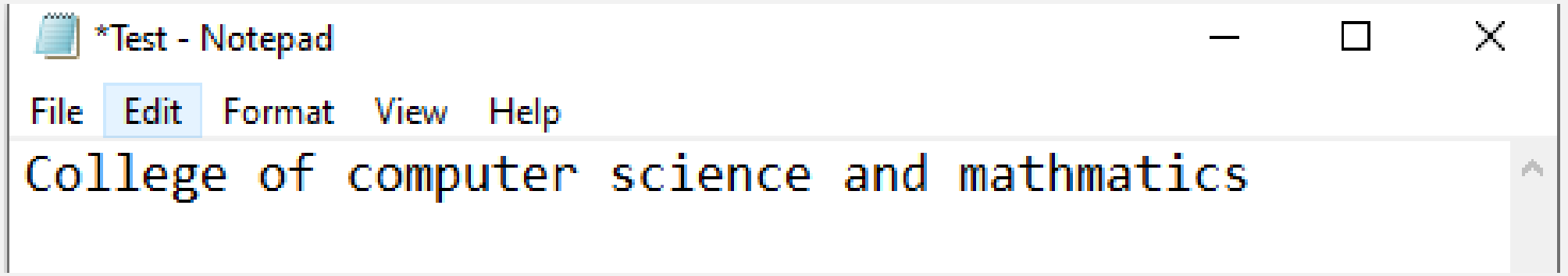
```
ADD
AX,CX
Lk:
BB:MOV
AX;1'com
CC:
JC
Count:
Press any key to continue .
```

FSTREAM EXAMPLES "OFSTREAM"

```cpp
#include <fstream>
#include <iostream>
using namespace std;

int main () {
   char data[100];

   // open a file in write mode.
   ofstream outfile;
   outfile.open("Test.txt");

   cout << "Writing something to the file" << endl;
   cin.getline(data, 100); // reads a line of text from the stream into the string.
   outfile << data << endl;
   outfile.close();
   return 0;
}
```

```
Writing something to the file
College of computer science and mathmatics
```

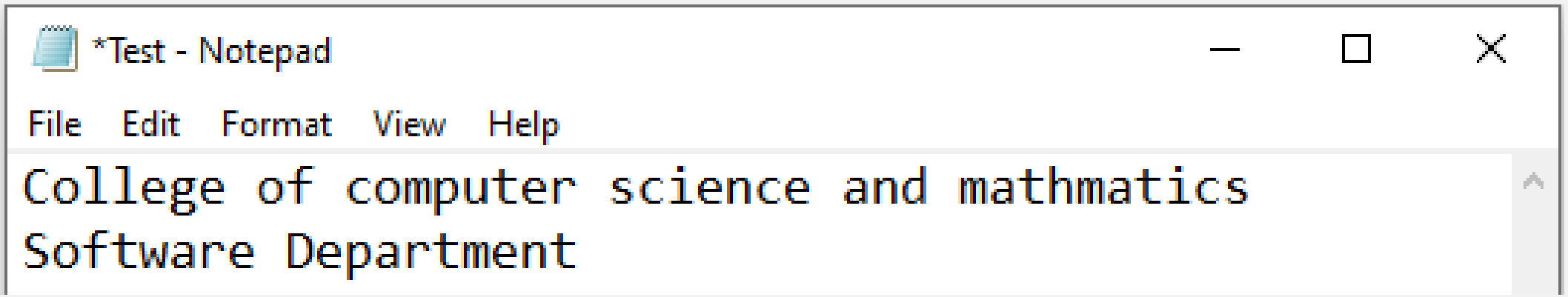
After execution:



FSTREAM EXAMPLES "OFSTREAM"

```cpp
#include <fstream>
#include <iostream>
using namespace std;

int main () {
   char data[100];

   // open a file in write mode.
   ofstream outfile;
   outfile.open("Test.txt", ios::app);

   cout << "Writing something to the file" << endl;
   cin.getline(data, 100); // reads a line of text from the stream into the string.
   outfile << data << endl;
   outfile.close();
   return 0;
}
```

```
Writing something to the file
Software Department
```

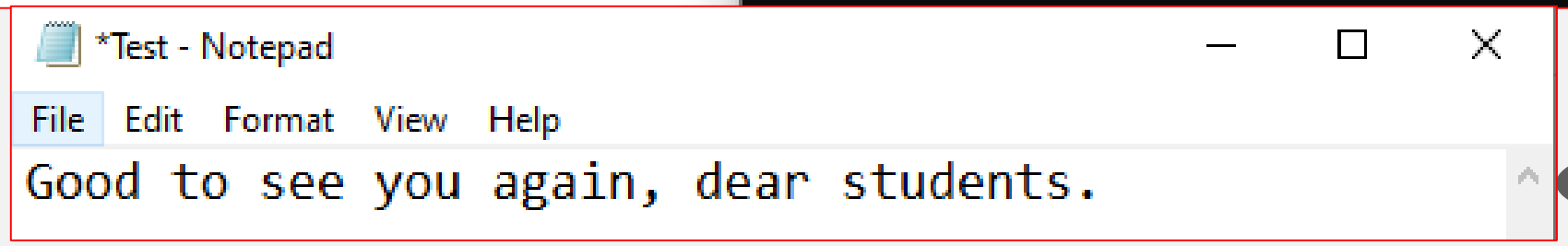


FSTREAM EXAMPLES "FSTREAM"

```cpp
#include <fstream>
#include <iostream>
using namespace std;

int main () {
    char data[100];
    fstream file;
    file.open("Test.txt", ios::in | ios::out);
    cin.getline(data,100);
    file << data << endl;
    //file >> data;
    //cout<< data <<endl;
    file.close();
    system("pause");
    return 0;
}
```

```
Good to see you again, dear students.
```

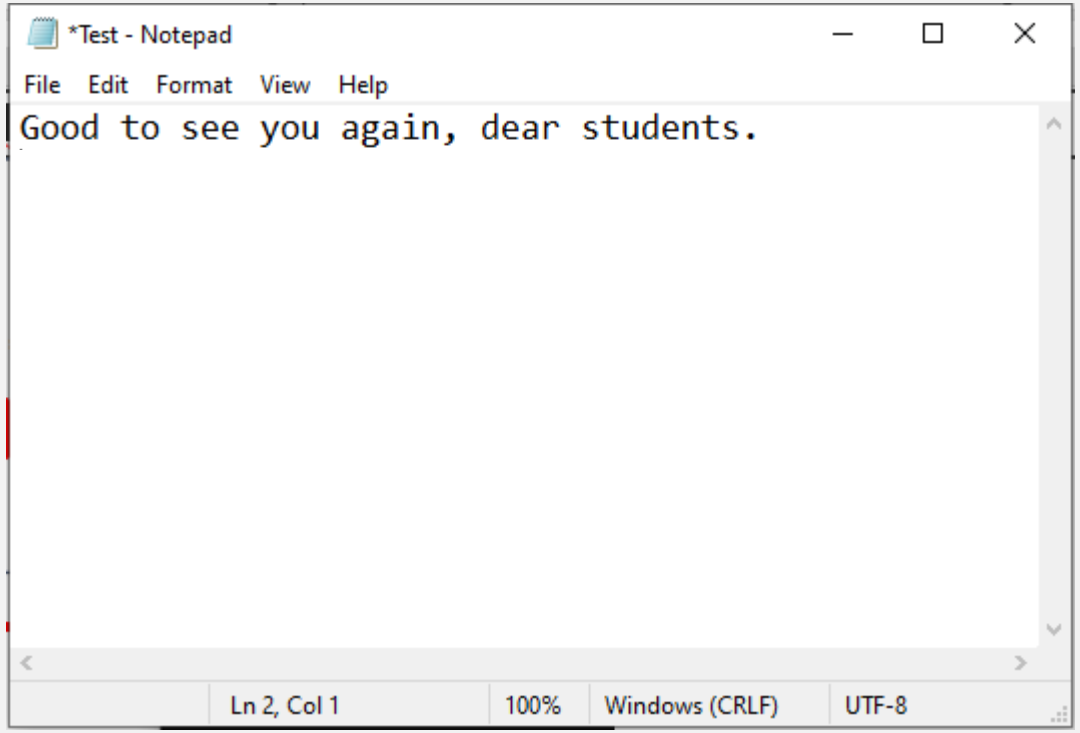
After execution:



FSTREAM EXAMPLES "FSTREAM"

```cpp
#include <fstream>
#include <iostream>
using namespace std;

int main () {
    char data[100];
    fstream file;
    file.open("Test.txt", ios::in | ios::out);
    //cin.getline(data,100);
    //file << data << endl;
    file >> data;
    cout<< data <<endl;
    file.close();
    system("pause");
    return 0;
}
```

```
Good
Press any key to continue . . .
```




```cpp
#include <fstream>
#include <iostream>
#include<string>
using namespace std;

int main () {
    string data;
    //char data[100];
    fstream file;
    file.open("Test.txt", ios::in | ios::out);
    //cin.getline(data,100);
    //file << data << endl;

    //file >> data;
    getline(file, data);
    cout<< data <<endl;
    file.close();
    system("pause");
    return 0;
}
```

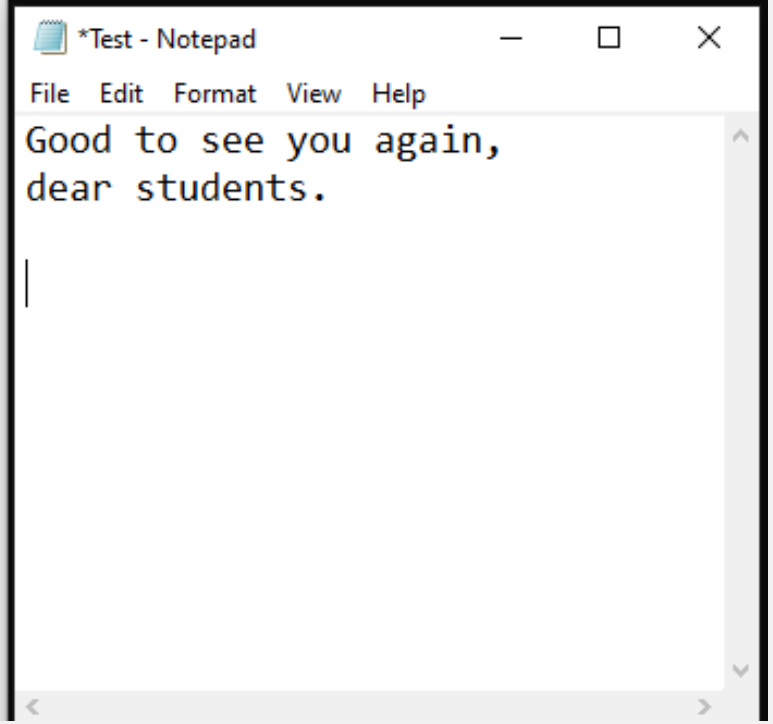
```
Good to see you again, dear students.
Press any key to continue . . .
```

FSTREAM EXAMPLES "FSTREAM"

```cpp
// reading a text file
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main () {
  string data;
  fstream file ("Test.txt", ios::in);
  while ( getline (file,data) ) // To read multiple lines inside the file
    {
      cout << data << '\n';
    }
    file.close();
    system("pause");
      return 0;
}
```



```
Good to see you again,
dear students.
Press any key to continue . . .
```



SETW FUNCTION

- setw() function helps in ***setting the field width*** which will be used on output operations.
- Syntax: setw(num)

Note: The following ***input-output manipulation*** library is applied to which **setw()** function belongs: #include <iomanip>

```
#include <iostream>
#include <iomanip>
using namespace std;
int main () {
cout << setw(10);
cout <<"Computer"<< endl;
system("pause");
return 0;
}
```

The output:
_ _ Computer

Since the string is **EIGHT** characters long, the first two places will be blank. The last **Eight** places are filled with the output.

SETW FUNCTION “CONTINUE’

Another example:

```
#include <iostream>
#include <iomanip>
using namespace std;
int main () {
const int Count = 3;
const int width = 4;
int row;
int i;
for(row=1;row<=5;row++)
{
for(i = 1; i <= Count; i++)
{
cout << setw(width) << row * i;
}
cout << endl;
}
system("pause");
return 0;
}
```

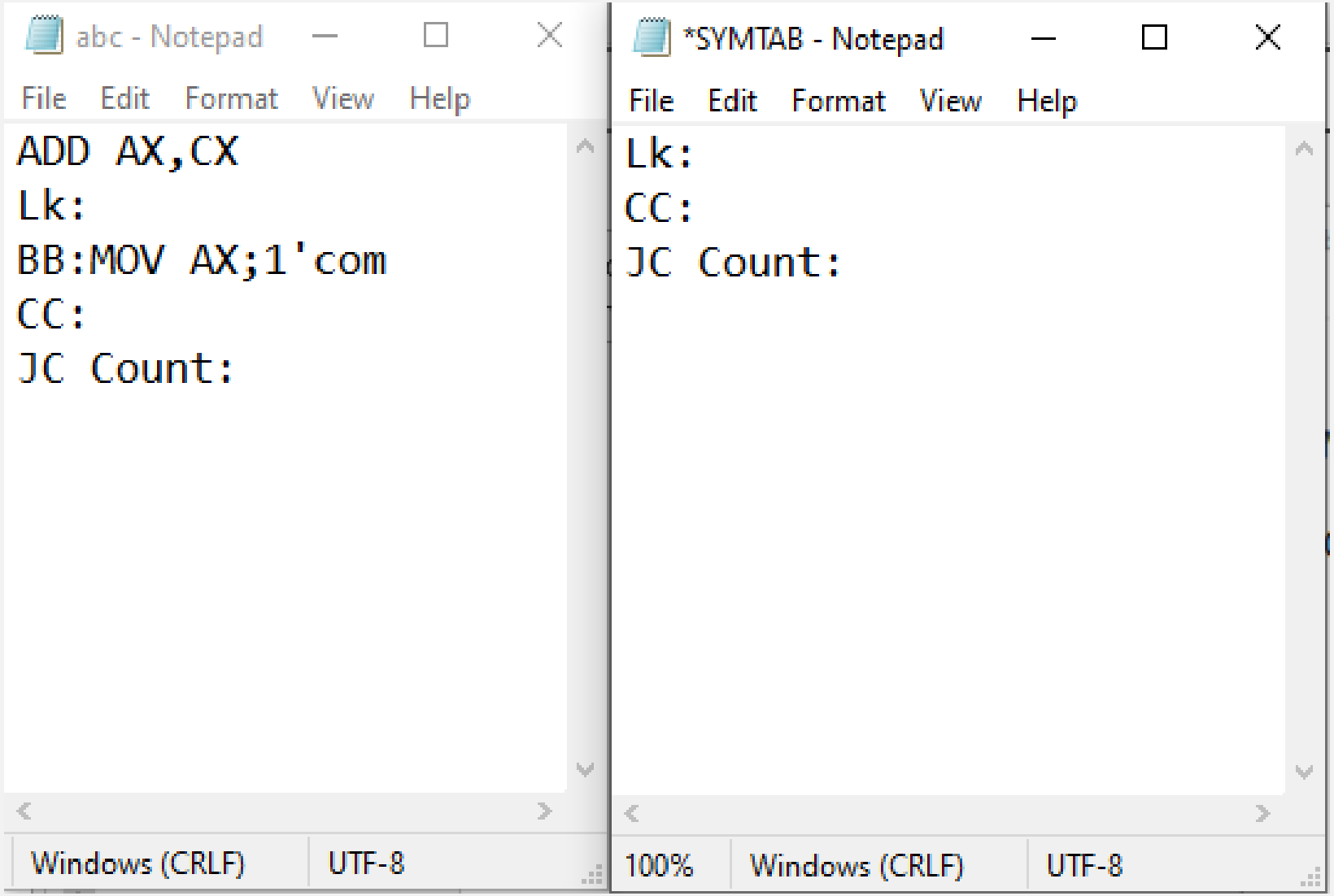
```
   1   2   3
   2   4   6
   3   6   9
   4   8  12
   5  10  15
Press any key to continue . . .
```

140 %

Locals

EXAMPLE

Q: Find labels in a file "abc.txt“, and save them into another file called "SYMTAB".



ANSWER

```
#include<iostream>
#include<fstream>
#include<string>
using namespace std;
void main(){

    string str;
    int i=0,j=0;
    ifstream infile;
    infile.open("abc.txt");
    ofstream outfile;
    outfile.open("SYMTAB.txt");

    if(infile.fail()) {
        cout<<"Error in file";
        exit(1);
    }
    while(!infile.eof()) {
        getline(infile, str);
         if( str.back() ==':') //  string.back() to get a reference to the last character in the string.
            {
                cout<< str << endl;
                outfile<< str << endl ;
         }
            }
    system("pause");
}
```

THE END