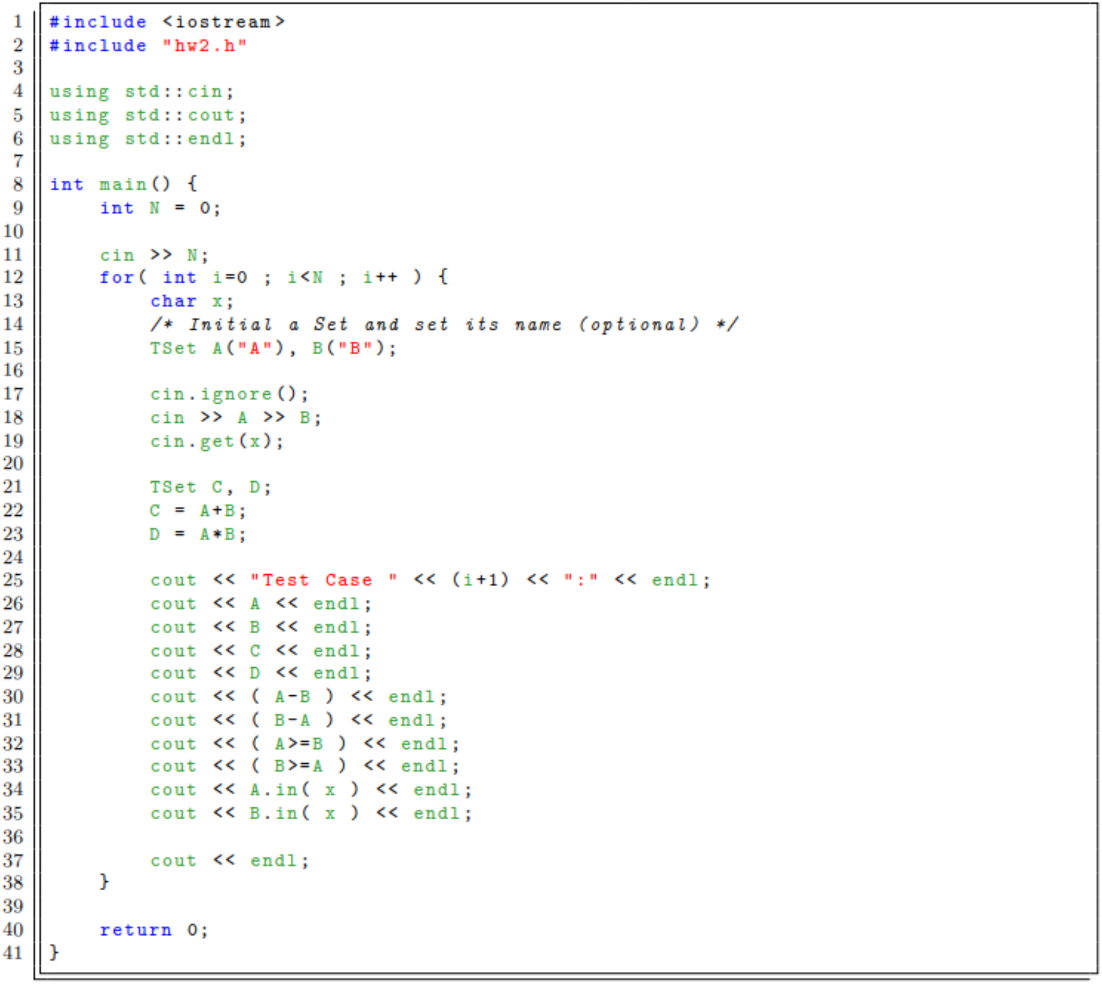
Data Structures

Note for Assignment #2

- The following example is written by TA. You don’t have to write so complicated as TA，but line21 to line 23 is necessary.

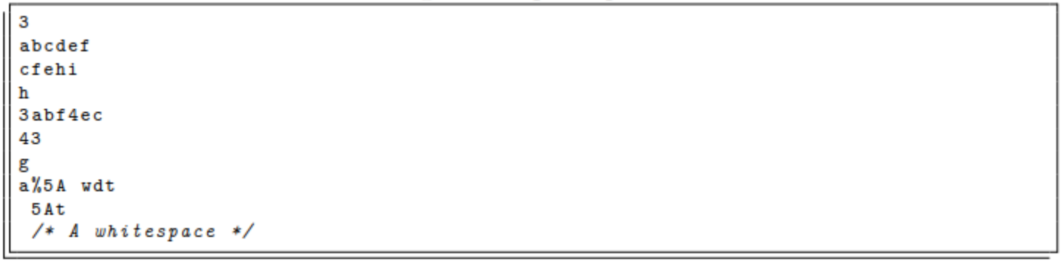
Listing 1: main.cpp



```
 1  #include <iostream>
 2  #include "hw2.h"
 3
 4  using std::cin;
 5  using std::cout;
 6  using std::endl;
 7
 8  int main() {
 9      int N = 0;
10
11      cin >> N;
12      for( int i=0 ; i<N ; i++ ) {
13          char x;
14          /* Initial a Set and set its name (optional) */
15          TSet A("A"), B("B");
16
17          cin.ignore();
18          cin >> A >> B;
19          cin.get(x);
20
21          TSet C, D;
22          C = A+B;
23          D = A*B;
24
25          cout << "Test Case " << (i+1) << ":" << endl;
26          cout << A << endl;
27          cout << B << endl;
28          cout << C << endl;
29          cout << D << endl;
30          cout << ( A-B ) << endl;
31          cout << ( B-A ) << endl;
32          cout << ( A>=B ) << endl;
33          cout << ( B>=A ) << endl;
34          cout << A.in( x ) << endl;
35          cout << B.in( x ) << endl;
36
37          cout << endl;
38      }
39
40      return 0;
41  }
```

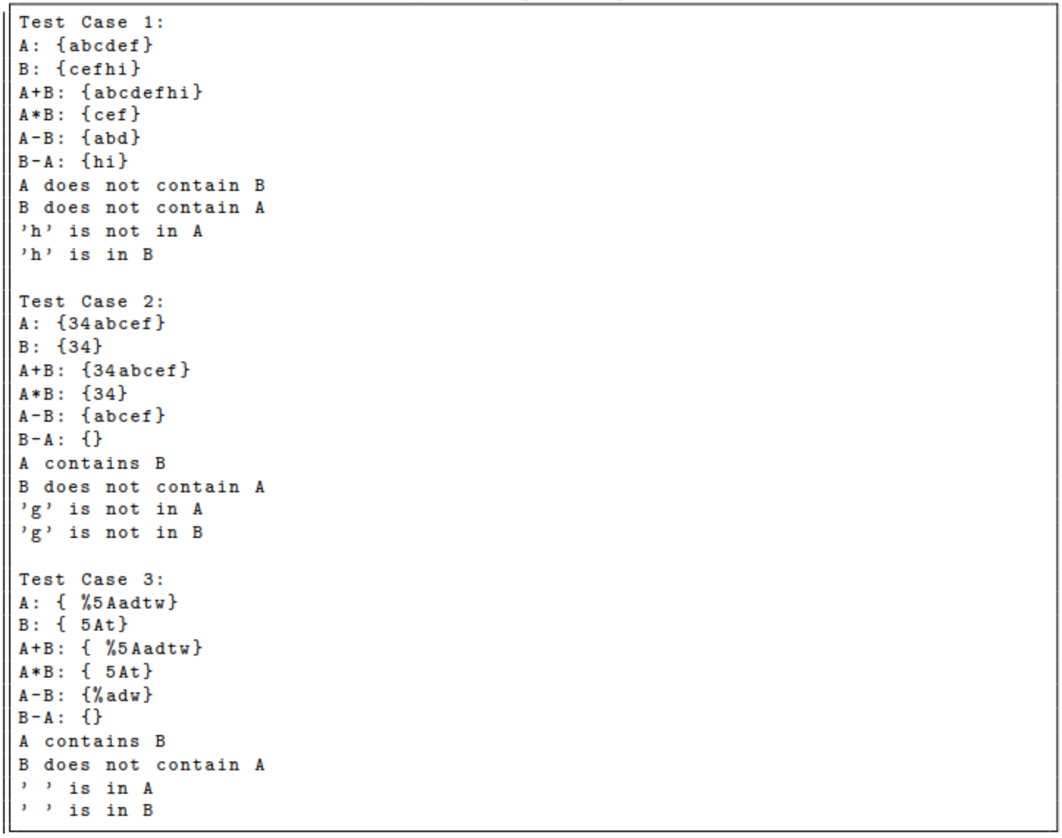
- The range of test data is for all characters on the keyboard (it means if you can’t output from your keyboard, it won’t appear.).You don’t have to consider correctness of the input data. The number of case and length of the test data is not fixed.

Listing 2: Sample Input



```
3
abcdef
cfehi
h
3abf4ec
43
g
a%5A wdt
 5At
 /* A whitespace */
```

Listing 3: Sample Output



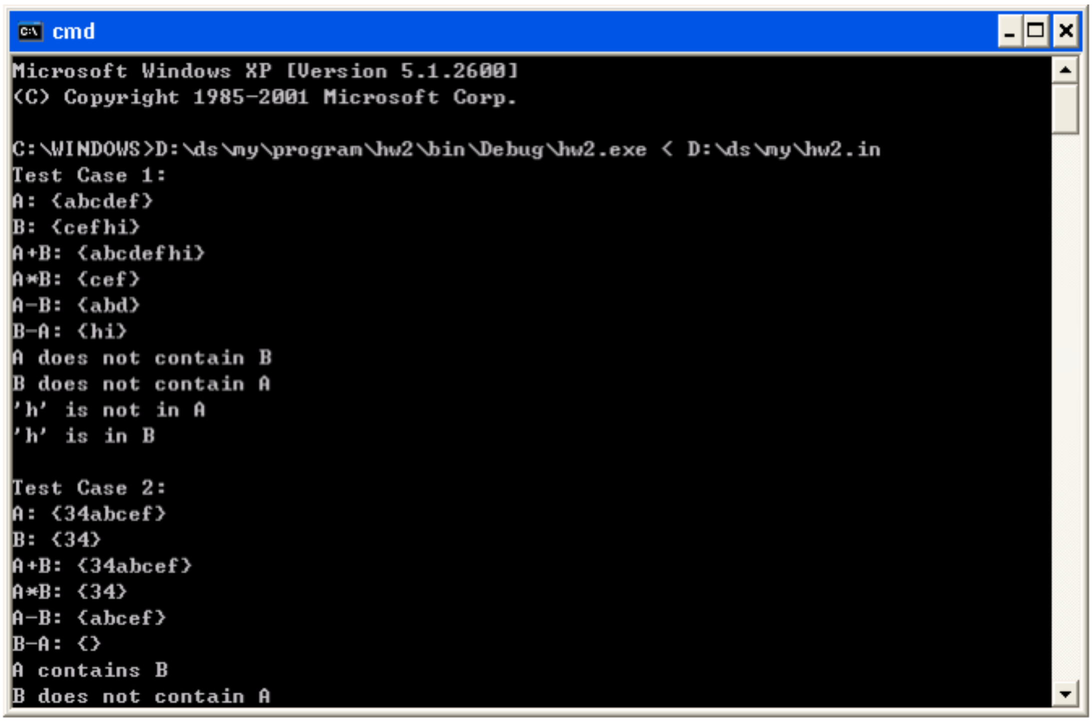
```
Test Case 1:
A: {abcdef}
B: {cefhi}
A+B: {abcdefhi}
A*B: {cef}
A-B: {abd}
B-A: {hi}
A does not contain B
B does not contain A
’h’ is not in A
’h’ is in B

Test Case 2:
A: {34abcef}
B: {34}
A+B: {34abcef}
A*B: {34}
A-B: {abcef}
B-A: {}
A contains B
B does not contain A
’g’ is not in A
’g’ is not in B

Test Case 3:
A: { %5Aadtw}
B: { 5At}
A+B: { %5Aadtw}
A*B: { 5At}
A-B: {%adw}
B-A: {}
A contains B
B does not contain A
’ ’ is in A
’ ’ is in B
```

- Some students ask the questions whether they need to read all the data simultaneously in order to output the same as in this example, or not ?
  Answer: NO!

- So how can I do it?
  Easily, it is to use the pipeline system to handle, as shown below:



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS>D:\ds\my\program\hw2\bin\Debug\hw2.exe < D:\ds\my\hw2.in
Test Case 1:
A: {abcdef}
B: {cefhi}
A+B: {abcdefhi}
A*B: {cef}
A-B: {abd}
B-A: {hi}
A does not contain B
B does not contain A
'h' is not in A
'h' is in B

Test Case 2:
A: {34abcef}
B: {34}
A+B: {34abcef}
A*B: {34}
A-B: {abcef}
B-A: {}
A contains B
B does not contain A
```

- We use the "<" so as to import the input data to a program. Relatively, you also can use ">" so as to export the output text to a file. This is a simple method for generating file (.out). I hope that it's helpful for you.