Write a program which uses a linked-list to maintain the terms of a polynomial. Each term of the polynomial will be inserted in a single node. Each node has three members, (Coefficient, Exponent and a pointer to the next node). (optionally, you can include a 4th member which would hold the sign for the coefficient.)

```
+--------+ +--------+ +--------+ +--------+ +--------+
Poly1-> | | +|--->| | |+|--->| | |+|--->| | |$|
+--------+ +--------+ +--------+ +--------+ +--------+
  ^    ^
 _|   |___Exponent
Coefficient
```

To begin with, we need a class that will hold the notion of a term.

```cpp
class polyTerm {
public:
    char sign;
    int coefficient;
    int exponent;
    polyTerm *next;  // Next term
polyTerm(char the_Sign, int theCoeff, int theExpo)
    void print(void);
};
```

Test the above class by creating a few ployTerm objects and printing them to make sure information is properly stored.

Once the above class is created and tested we can proceed to create a new class called polynomial. The specification of this class is given below. This new class will use the previously defined class called polyTerm in its definition. (see below).

```cpp
class polynomial {
private:
    polyTerm *polyHead;
public:
    polynomial() {polyHead = NULL;}
    void insertAtBeginning(char the_Sign, int theCoeff, int theExpo);
    void insertAtEnd(char theSign, int theCoeff, int theExpo);
    void insertInOrder(char theSign, int theCoeff, int theExpo);
    void print(void);
};
```

This class allows its clients to create a polynomial, by first instantiating an object of type “polynomial”, then inserting one or more terms in it. For example:

```cpp
polynomial poly1;
poly1.insertAtBeginning('+', 3, 2);
```
poly1.insertAtBeginning('-', 8, 7);
poly1.insertAtBeginning('+', 1, 5);
poly1.print();

Make sure to develop a good test plan and test each function of the class properly.

For the polynomial to be properly ordered, you will need to insert the terms in sorted order (based on the EXPONENT of the term). If the term being inserted, has the same exponent (as a term in the linked list) add them together! (don't create a new node)

**Sample input:**

\[
+ 3 \times 2 - 8 \times 7 + 1 \times 5 - 2 \times 2 + 1 \times 1 - 7 \times 0 + 4 \times 3
\]

The format of each term is as follows:
- Operator (+ or -)
- Coefficient (integer number, one or more digits)
- X
- Exponent (integer number, one or more digits)

**Sample output:**

\[
- 8 \times 7 + 1 \times 5 + 4 \times 3 + 1 \times 2 + 1 \times 1 - 7 \times 0
\]

**What to hand in:**

1) Printed Source code (documented, formatted)
2) Screen shots testing your code. Go beyond what I have listed above. (But the first test case should be the sample input provided)
3) Zipped project, as well as the screen shots (Uploaded into canvas)
void main()
{
    polynomial poly4;
    cout << "INPUT: + 3 x 2 - 8 x 7 + 1 x 5 - 2 x 2 + 1 x 1 - 7 x 0 + 4 x 3" << endl;
    poly4.insertInOrder('+', 3, 2); poly4.print(); system("pause");
    poly4.insertInOrder('-', 8, 7); poly4.print(); system("pause");
    poly4.insertInOrder('+', 1, 5); poly4.print(); system("pause");
    poly4.insertInOrder('-', 2, 2); poly4.print(); system("pause");
    poly4.insertInOrder('+', 1, 1); poly4.print(); system("pause");
    poly4.insertInOrder('-', 7, 0); poly4.print(); system("pause");
    poly4.insertInOrder('+', 4, 3); poly4.print(); system("pause");
    cout << "OUTPUT: - 8 x 7 + 1 x 5 + 4 x 3 + 1 x 2 + 1 x 1 - 7 x 0 " << endl;
}