

## EE2410 Data Structure Coding HW #1 (Chapter 1~2 of textbook)

**due date 4/2/2024 23:59**

You should submit:

- (a) All your source codes (C++ file).
- (b) Show the execution trace of your program, i.e., write a client main() to demonstrate all functions you designed using example data.

Submit your homework before the deadline (midnight of 4/2). Fail to comply (**late** homework) will have **ZERO score**. **Copy** homework will have **SERIOUS consequences**.

**Arrays: due date: 23:59, 4/2/2024 (Tue.)**

1. (30%)

Write a C++ program to implement the **ADT2.3 Polynomial** below using Representation 3 (dynamic array of (coef, exp) tuples).

```
class Polynomial;
class Term{
friend Polynomial;
private:
    float coef;
    int exp;
};
class Polynomial {
//  $p(x) = a_0 x^{e_0} + \dots + a_n x^{e_n}$ 
// where  $a_i$  is nonzero float and  $e_i$  is non-negative int
public:
    Polynomial();
    //construct the polynomial  $p(x) = 0$ 
    ~Polynomial();
    //destructor
    Polynomial Add(Polynomial poly);
    //return the sum of *this and poly
    Polynomial Subt(Polynomial poly);
    //return the difference of *this and poly
    Polynomial Mult(Polynomial poly);
    //return the product of *this and poly
    void NewTerm(const float theCoeff, const int theExp);
```

```

float Eval(float f);
//Evaluate the polynomial *this at f and return the results
int operator!();
    // if *this is the zero polynomial, return 1; else return 0;
float Coef(int e);
    // return the coefficient of e in *this
int LeadExp();
    // return the largest exponent in *this
friend ostream& operator<<(ostream& os, Polynomial& p);
friend istream& operator>>(istream& is, Polynomial& p);
private:
    Term* termArray;
    int capacity;
    int terms;
};

```

Show the results of one runs of your program (execution trace) to **demonstrate** all the operations (Add, Subt, Mult, Eval, !, Coef, and LeadExp) as well as input, output functions as follows:

```

int main(){
    construct polynomial objects a, b
    use >> to build polynomial object  $a = 2x^3 + 3x^2 + 4x + 5$ ,  $b = x^3 - x^2 + x - 1$ 
    demo <<
    demo << results of Add, Subt, Mul
    demo results of a.Exal(1), b.Eval(2), a.Coef(5), b.LeadExp
}

```

```

D:\C++\data_structure\hw1\finished_code\q1.exe
Input Polynomial a:
Please input in the form {(coef,exp),(coef,exp),...} : {(2,3),(3,2),(4,1),(5,0)}
Input Polynomial b:
Please input in the form {(coef,exp),(coef,exp),...} : {(1,3),(-1,2),(1,1),(-1,0)}
Polynomial a = 2x^3 + 3x^2 + 4x + 5
Polynomial b = x^3 - x^2 + x - 1
a + b = 3x^3 + 2x^2 + 5x + 4
a - b = x^3 + 4x^2 + 3x + 6
a * b = 2x^6 + x^5 + 3x^4 + 2x^3 - 4x^2 + x - 5
a(x = 1) = 14
b(x = 2) = 5
a.Coeff(5) = 0
b.LeadExp = 3

Process returned 0 (0x0)   execution time : 18.910 s
Press any key to continue.

```

The program is designed to take in input polynomials in descending order.

## 2. (35%)

Write a C++ program to implement the **ADT2.4 SparseMatrix** in textbook shown below.

```
class SparseMatrix
```

```
//三元組，<列，行，值>，的集合，其中列與行為非負整數，
```

```
//並且它的組合是唯一的；值也是個整數。
```

```
public:
```

```
    SparseMatrix(int r, int c, int t);
```

```
    //constructor.
```

```
    //r is #row, c is #col, t is #non-zero terms
```

```
    SparseMatrix Transpose( );
```

```
    SparseMatrix FastTranspose( );
```

```
    //回傳將 *this 中每個三元組的行與列交換後的 SparseMatrix
```

```
    SparseMatrix Add(SparseMatrix b);
```

```
    // 如果 *this 和 b 的維度一樣，那麼就把相對應的項給相加，
```

```
    // 亦即，具有相同列和行的值會被回傳；否則的話丟出例外。
```

```
    SparseMatrix Multiply(SparseMatrix b);
```

```
    // 如果*this 中的行數和 b 中的列數一樣多的話，那麼回傳的矩陣 d= *this 和 b
```

```
    //（依據  $d[i][j] = \sum(a[i][k] \cdot b[k][j])$ ，其中  $d[i][j]$  是第  $(i,j)$  個元素）相乘的結果。
```

```
    // k 的範圍從 0 到*this 的行數減 1；如果不一樣多的話，那麼就丟出例外。
```

```
    //other needed functions
```

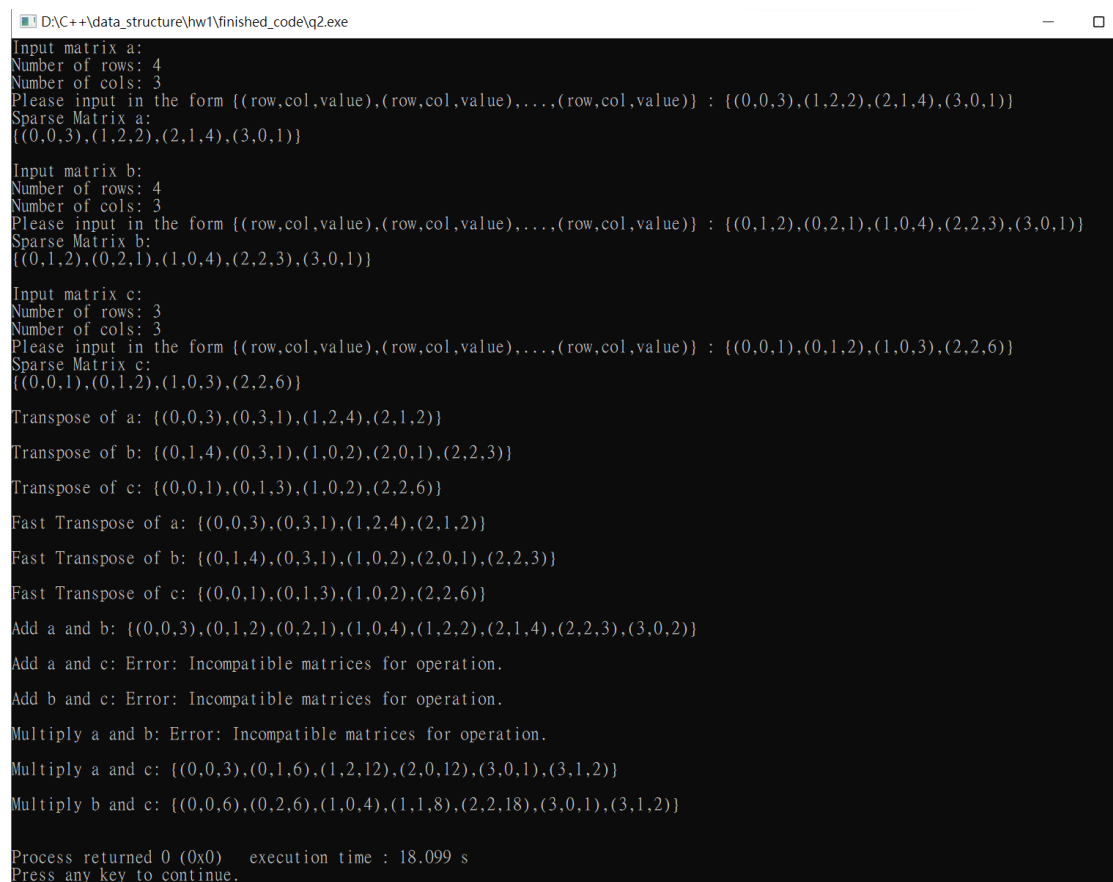
```
};
```

You should build you program based on the example codes in the book and

implement the **Add** function and functions to **input**, **output** a sparse matrix by **overloading** the **>>** and **<<** operators.

You should try out at least one runs of your program to demonstrate the Add, Mult, FastTranspose, and input, output functions.

```
int main(){
    use >> to build sm object a(4x3, 4 terms), b(4x3, 5 terms), c(3x3, 4 terms)
    demo <<
    demo << results of Transpose, Fast Transpose, Add, Mul
}
```



```
D:\C++\data_structure\hw1\finished_code\q2.exe
Input matrix a:
Number of rows: 4
Number of cols: 3
Please input in the form {(row,col,value),(row,col,value),...,(row,col,value)} : {(0,0,3),(1,2,2),(2,1,4),(3,0,1)}
Sparse Matrix a:
{(0,0,3),(1,2,2),(2,1,4),(3,0,1)}

Input matrix b:
Number of rows: 4
Number of cols: 3
Please input in the form {(row,col,value),(row,col,value),...,(row,col,value)} : {(0,1,2),(0,2,1),(1,0,4),(2,2,3),(3,0,1)}
Sparse Matrix b:
{(0,1,2),(0,2,1),(1,0,4),(2,2,3),(3,0,1)}

Input matrix c:
Number of rows: 3
Number of cols: 3
Please input in the form {(row,col,value),(row,col,value),...,(row,col,value)} : {(0,0,1),(0,1,2),(1,0,3),(2,2,6)}
Sparse Matrix c:
{(0,0,1),(0,1,2),(1,0,3),(2,2,6)}

Transpose of a: {(0,0,3),(0,3,1),(1,2,4),(2,1,2)}
Transpose of b: {(0,1,4),(0,3,1),(1,0,2),(2,0,1),(2,2,3)}
Transpose of c: {(0,0,1),(0,1,3),(1,0,2),(2,2,6)}

Fast Transpose of a: {(0,0,3),(0,3,1),(1,2,4),(2,1,2)}
Fast Transpose of b: {(0,1,4),(0,3,1),(1,0,2),(2,0,1),(2,2,3)}
Fast Transpose of c: {(0,0,1),(0,1,3),(1,0,2),(2,2,6)}

Add a and b: {(0,0,3),(0,1,2),(0,2,1),(1,0,4),(1,2,2),(2,1,4),(2,2,3),(3,0,2)}
Add a and c: Error: Incompatible matrices for operation.
Add b and c: Error: Incompatible matrices for operation.
Multiply a and b: Error: Incompatible matrices for operation.
Multiply a and c: {(0,0,3),(0,1,6),(1,2,12),(2,0,12),(3,0,1),(3,1,2)}
Multiply b and c: {(0,0,6),(0,2,6),(1,0,4),(1,1,8),(2,2,18),(3,0,1),(3,1,2)}

Process returned 0 (0x0)   execution time : 18.099 s
Press any key to continue.
```

The program will throw exception when encountering incompatible matrix operations.

Terms in a sparse matrix are arranged in row-ascending order.

3. (35%)

Write a C++ program to implement the **ADT2.5 String**.

```
class String
{
```

**public:**

```
String(char *init, int m);  
// constructor using input string init of length m  
bool operator == (String t); //equality test  
bool operator ! ( ); // empty test, true or false  
int Length( ); //get the number of characters of *this  
String Concat(String t);  
// concatenation with another string t  
String Substr(int i, int j); // generate a substring  $i \sim i+j-1$   
int Find(String pat);  
int FastFind(String pat);  
// Return an index  $i$  such that pat matches the substring  
// of the object begins at position  $i$ . Return -1 if pat  
// is empty or not a substring of the object  
String Delete(int start, int length); //remove length characters beginning  
// at start  
String CharDelete(char c); //returns the string with all occurrence of c  
//removed.  
int Compare(String y); //compare two strings of letters of alphabet.  
//return -1 if  $<y$ , 0 if  $=y$ , and 1 if  $>y$ .  
//If two strings of letter of alphabet,  $x = (x_0, \dots, x_{m-1})$  and  $y = (y_0, \dots, y_{n-1})$   
//where  $x_i, y_j$  are letters, then the Compare member function will decide  
//whether  $x < y$ ,  $x = y$ , or  $x > y$ , where  $x < y$  means if  $x_i = y_i$  for  $0 \leq i < j$  and  $x_j < y_j$   
//or if  $x_i = y_i$  for  $0 \leq i \leq m$  and  $m < n$ .  $x = y$  means  $m = n$  and  $x_i = y_i$  for  $0 \leq i < n$ .  $x > y$   
//means if  $x_i = y_i$  for  $0 \leq i < j$  and  $x_j > y_j$  or if  $x_i = y_i$  for  $0 \leq i \leq n$  and  $m > n$ .  
}
```

You should try out at least two example runs of your program to demonstrate all those functions.

D:\C++\data\_structure\hw1\finished\_code\q3.exe

```
Length of the string: 20
Input the string: abacaabaccabacabaabb
Length of the string: 10
Input the string: HelloWorld

String s1 is: abacaabaccabacabaabb
with length = 20
String s2 is: HelloWorld
with length = 10

s1 is not equal to s2

s1 is not empty
s2 is not empty

s1.Concat(s2) is: abacaabaccabacabaabbHelloWorld
with length = 30

Input 'start' and 'length' for substring s1 (separate the two numbers by space): 2 5
s1.Substr(2, 5) = acaab
with length = 5

Input a pattern to find in s1:
Length of the string: 6
Input the string: abacab\
s1.Find(pat) = 10
s1.FastFind(pat) = 10

Which part do you want to delete in string s1 (input 'start' and 'length', separate the two numbers by space): 2 5
s1.Delete(2, 5) = abaccabacabaabb
with length = 15

Input a character to delete in string s1: b
s1.CharDelete(b) = aacaaccaacaaa
with length = 14

String s1 is larger than String s2

Process returned 0 (0x0)   execution time : 71.959 s
Press any key to continue.
```