Task

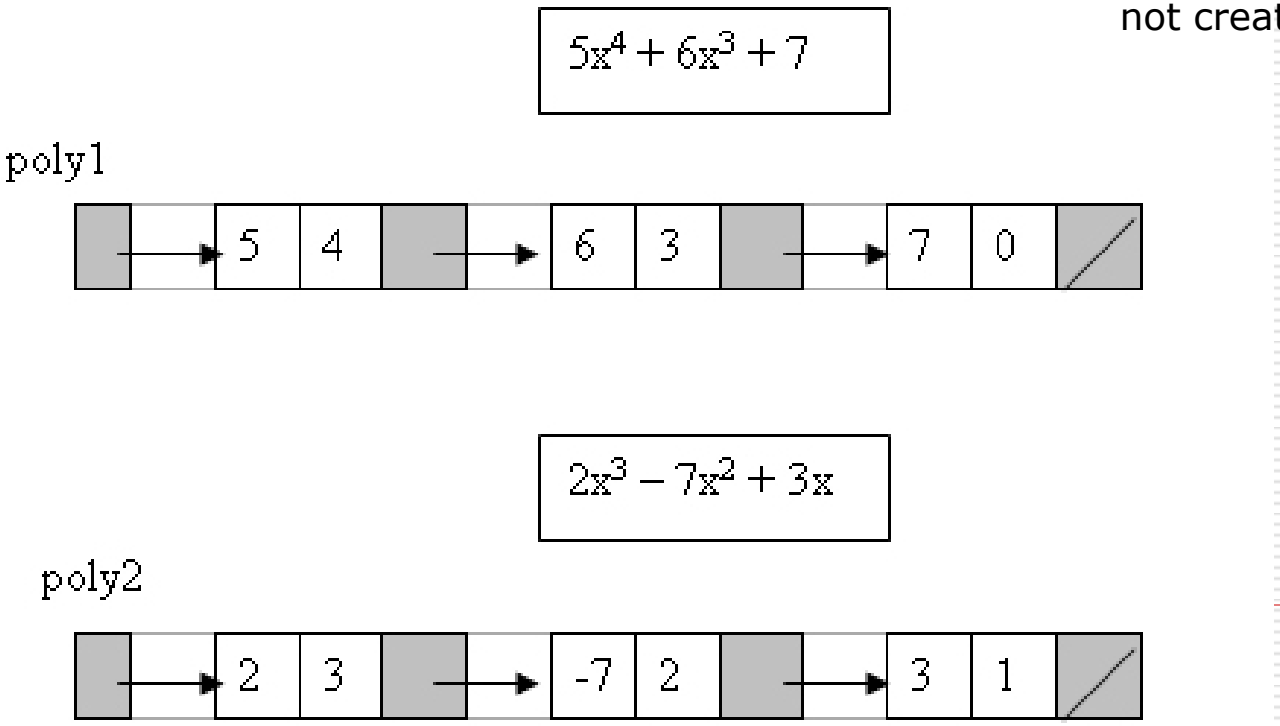
- The goal of this programming project is to implement a **polynomial calculator**, i.e., a C++ program that reads, prints, adds and multiplies polynomials.

- You have to
  - Implement the ADT Polynomial in C++
  - Write a main function that reads from input file, creates and manipulates objects of ADT Polynomial, and prints out the results.

System Requirements (1/5)

- **The ADT Polynomial.**
- Using a linked list to store the data (coefficient-exponent pairs, arranged in descending order by exponent).

If the coefficient is 0, do not create a node for it.



System Requirements (2/5)

- Operation Contract (specification) for the ADT Polynomial:

`createPolynomial()`

Creates a degree 0 polynomial with value 0, i.e., P(x)=0.

`destroyPolynomial()`

Destroys a polynomial.

`getDegree():integer {query}`

Returns the degree of the polynomial.

$(4x^4-3x+3)$

`getNonZero():integer {query}`

Returns the number of variables with non-zero coefficient.

`getPoly(): string {query}`

Returns a string (in the form of $c_n x^n+\ldots+c_1 x^1 + c_0$) representing the polynomial.

System Requirements (3/5)

set(in **n**:integer, in **c**:integer) **throw PolynomialException**

- Set the coefficient of the **n** degree variable to **c**.
- Throw exception if the operation fails

cx^n

- Hint: the **new** operator may thow bad_alloc exception

```
#include <cstddef>   // for NULL
#include <new>       // for bad_alloc
```

System Requirements (4/5)

add(in **poly**:polynomial) **throw PolynomialException**

- Add **poly** to the current polynomial, the content of **poly** should not be changed.
- Throw exception if the operation fails

```
cur := poly + cur
```

System Requirements (5/5)

multiplies(in **poly**:polynomial) **throw PolynomialException**

- Multiply **poly** with the current polynomial, the content of **poly** should not be changed.
- Throw exception if the operation fails

```
cur := poly * cur
```

Program Input & output (1/3)

- Input:
  - The **input file** is in the following format:

$(3x^2+4x-6)*(4x^4-3x+3)+(12x^3+5x^6-4)$

```
3 2
4 1
-6 0
*
4 4
-3 1
3 0
+
12 3
5 6
-4 0
```

$3x^2+4x-6$, polynomial 1

an operator

We assume that all the operators have the same precedence

$4x^4-3x+3$, polynomial 2

an operator

$12x^3+5x^6-4$, polynomial 3

Program Input & output (2/3)

- `%> poly input_file`
  - Your `main` program SHOULD receive parameter from the command line.

```
int main(int argc, char * argv[])
{
  // arvg[1] = the string of input_file
  ...
}
```

- In the `main` program, you SHOULD create objects of *ADT polynomial* according to the input file, do the corresponding operations (*add* or *multiply*), and at the end PRINT OUT the result in the required format.

Program Input & output (3/3)

- Output format:
- $(3x^2+4x-6)*(4x^4-3x+3)+(12x^3+5x^6-4)$

```
Polynomial 1:3x^2+4x-6
Degree:2
# of nonzero var: 2

Polynomial 2:4x^4-3x+3
Degree:4
# of nonzero var: 2

Polynomial 3:5x^6+12x^3-4
Degree:6
# of nonzero var: 2

Result Polynomial: 17x^6 + 16x^5 -24x^4 +3x^3 -3x^2 +30x -22
Degree:6
# of nonzero var: 6
```