OOP (1010), Summer 2015

Lab 4

Please implement the following algorithm to compress or decompress a file of n bytes.

```
let i = 0;
while( i < n )
   let x be an 8-bit signed integer;
   x = read the i-th byte from F.
   if  (0 ≤ x ≤ 127){
       read the next x+1 bytes from F.
       i = i + x + 2;
   }
   else if (-127 ≤ x ≤ -1){
       read the next byte -x+1 times.
       i = i + 2
   }
}
```

For example, given a compressed file of 15 bytes, the value of each byte listed as follows:

```
FE AA 02 80 00 2A FD AA 03 80 00 2A 22 F7 AA
```

Then the original bytes are

```
AA AA AA 80 00 2A AA AA AA AA 80 00 2A 22 AA AA AA AA AA AA AA AA AA
AA
```

Input / Output

Your program should be able to continuously read input lines from `cin` until EOF is detected. Each line consists of three parts separated by white spaces are as follows:

command: A single character. **'C'** represents compress; **'D'** represents decompress.
source: The path of input file,
output: The Path of output file.

For example,

C D:/test/A.bmp D:/test/A.dat

means that compressing D:/test/A.bmp and output to D:/test/A.dat.

After a line is processed including compressing / decompressing and file storage, please

show the size of input file, the size of output file, and the processing time in thousandths of a second (ex: 1.234 sec).

Requirements

1. All file operations must be done by **`fstream`**.
2. All data array must be **dynamic**, allocated by **`new`** operation, and managed by **`std::unique_ptr`**.