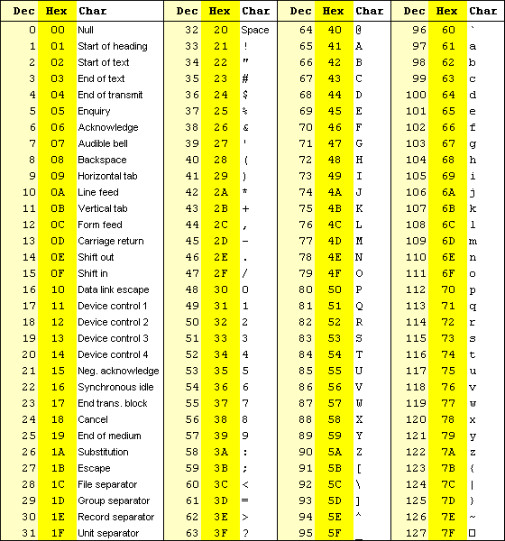
**<COMP\_HW2>**

2015-18634

Hyunjong Lee

**[Problem 1]**

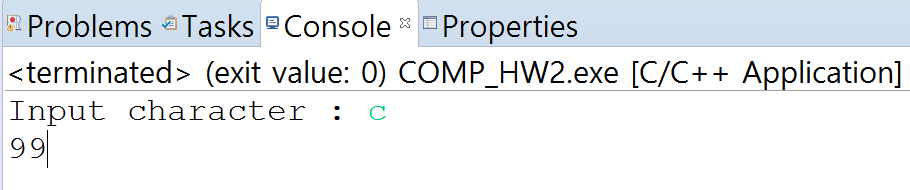
I used ‘(std::)cin’ keyword to get input character. Then I printed out the corresponding interger value of a character in ASCII code table by using ‘static\_cast<int>’. The ‘static\_cast<type>(expression)’ does implicit conversion between types such as int, float, char, double and so on.



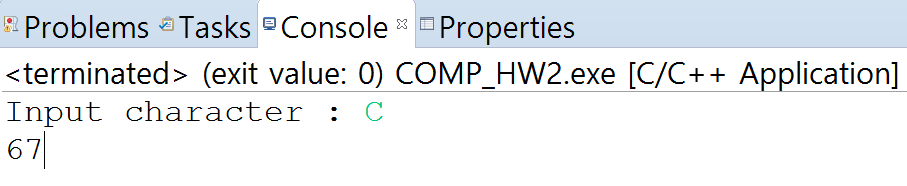
[Figure ] ASCII Code Table

* Execution Result

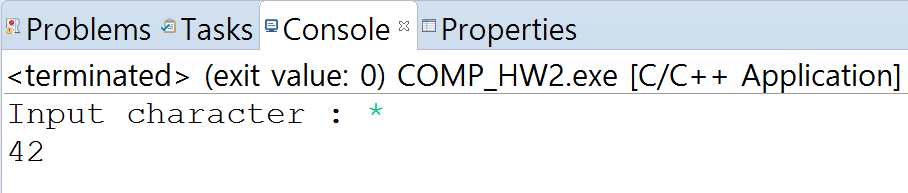
(1)



(2)



(3)



**[Problem 2]**

When input is between 10000 and 99999 (= input is 5-digit number), the program prints its individual digits separated by three spaces.

1st digit : input / 10000 → 42339 / 10000 = 4

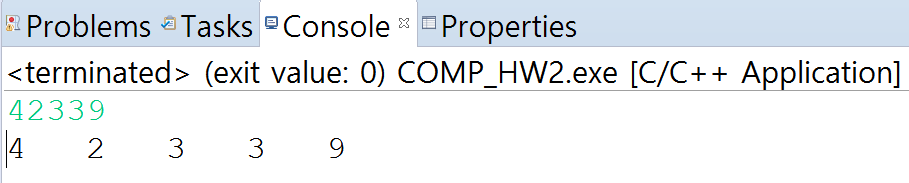
2nd digit : (input / 1000) % 10 → (42339/1000) % 10 = 42 % 10 = 2

3rd digit : (input / 100) % 10 → (42339/100) % 10 = 423 % 10 = 3

4th digit : (input / 10) % 10 → (42339/10) % 10 = 4233 % 10 = 3

5th digit : input % 10 → 42339 % 10 = 9

* Execution Result



**[Problem 3]**

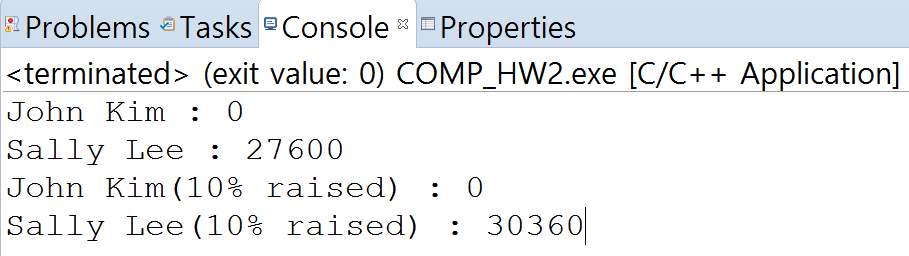
· If the monthly salary is not positive, I set it to 0 by using if statement.

( if(monthly salary <= 0) { monthly salary substitutes to 0; } )

· I make two employee objects with constructor. (Employee employee1(string, string, int); )

· I used setter and getter to raise the monthly salary. I set new monthly salary to 110% of original monthly salary. Then, I multiply 12 to calculate yearly salary.

* Execution Result



**[Problem 4]**

The program printed exact value of the powers of the integer 2 until 2^30. After that, 2^31 printed -2147483648(-2^31), and then powers of 2 printed 0. This is because computer only uses 32bits to store integer value. The first bit is sign bit, and the rest bits are for absolute value. So program could print power of 2 from 2^1 to 2^30. (2^30 = )

2^31 = means sign bit is 1 and absolute value is 0. So it is printed as -2^31 because computer use 2’s complement to express negative integer.

After 2^31, overflow occurred. Thus, computer think powers of 2 bigger than 2^32 (= ) as .

**[Problem 5]**

First, I define constant product1 to product5 that represents each product’s price.

Second, I made array that i-th element represents (i+1) product’s quantity sold. ( i=0,1,2,3,4 ) I initialized all elements to 0.

Third, I got character input for product number to handling exit. If input is ‘x’ or ‘X’, terminate program. While input is neither ‘x’ nor ‘X’, I also get input for quantity sold of that product. Then if product number is between 1 and 5, I increase corresponding product’s quantity by using switch statement.

After input, I calculate total price.

* Execution Result

