

Assignment #1, Object Oriented Programming, Spring Semester, 2024

Due date: 2024.03.29

1. Write a program that reads 5 integers and prints the minimum, maximum, and average values. (If a decimal part occurs, it must be rounded.)

```
Enter the five numbers: 10 6 29 31 88
MIN: 6
MAX: 88
AVG: 33
```

2. Write a program that print a 2-dimension array (5 × 10) with numbers randomly generated and also print sum and average of each row with '|' as a separator. The random numbers should be integer from 1 to 99. The computed average value should be presented with an integer number. (If a decimal part occurs, it must be rounded.) (The spaces between numbers must be as follow: one space or two spaces or three spaces)

```
4 91 51 99 81 6 88 22 95 28 | 565 | 57
56 77 3 62 7 32 85 48 24 63 | 457 | 46
95 2 83 45 12 4 58 47 35 12 | 393 | 39
31 26 17 7 60 16 24 83 79 27 | 370 | 37
1 8 3 6 8 9 1 1 9 7 | 53 | 5
```

3. Write a program that reads 10 characters and prints the first and the last on one line, the second and the ninth on the next line, the third and seventh on the next line, and so forth. Sample input and the results are shown below.

```
Enter ten characters: a b c d e E D C B A
Input characters are:
a A
b B
c C
d D
e E
```

4. Write a function that takes a positive integer value and returns the number with its digits reversed. For example, given the number 8263, the function should return 3628.

```
Enter the number: 8263
Reversed number: 3628
```

Assignment #1, Object Oriented Programming, Spring Semester, 2024

Due date: 2024.03.29

```
Enter the number: 2000000001
Reversed number: 1000000002
```

```
Enter the number: 2630
Reversed number: 362
```

5. Write a program that prompts a user for an integer value in the range 0 to 32,767 and then prints the individual digits of the numbers on a line with two spaces between the digits. The first line is to start with the leftmost digit and print all five digits; the second line is to start with the second digit from the left and print four digits, and so forth. For example, if the user enters 1234, your program should print.

```
Enter the number: 1234
0  1  2  3  4
1  2  3  4
2  3  4
3  4
4
```

6. Write a program that reads a floating-point number and prints the ceiling, floor, and rounding values without using standard library functions in C and C++. This program calculates to two decimal places. (If the third decimal place is less than 5, the absolute value is rounded down. If the third decimal place is 5 or more than 5, the absolute value is rounded up.)

```
Enter the floating-point number: 12.234
Ceiling: 12.24
Floor: 12.23
Rounding: 12.23
```

```
Enter the floating-point number: 1.035
Ceiling: 1.04
Floor: 1.03
Rounding: 1.04
```

Assignment #1, Object Oriented Programming, Spring Semester, 2024

Due date: 2024.03.29

```
Enter the floating-point number: -1.035
Ceiling: -1.03
Floor: -1.04
Rounding: -1.04
```

7. Write a recursive function `power(base,exponent)` that, when invoked, returns $base^{exponent}$. For example, `power(3,4) = 3 × 3 × 3 × 3`. Assume that exponent is an integer greater than or equal to 0. Hint: The recursion step would use the relationship: $base^{exponent} = base \times base^{exponent-1}$. And the termination condition occurs when exponent is equal to 1 because $base^0 = 1$.

```
Enter the base: 3
Enter the exponent: 4
power(3,4): 81
```

8. The greatest common divisor of integers x and y is the largest integer that evenly divides both x and y . Write a recursive function `gcd` that returns the greatest common divisor of x and y , which is defined recursively as follows:
If y is equal to 0, then `gcd(x,y)` is x ; otherwise, `gcd(x,y)` is `gcd(y,x % y)`, where `%` is the modulus operator.

```
Enter the 1st number: 12
Enter the 2nd number: 18
gcd(12,18): 6
```

9. The least common multiple of integer x and y is the smallest positive integer that is a multiple of both x and y . Write a function `lcm(x,y)` that returns the least common multiple of x and y using `gcd(x,y)` function of the problem 8.

```
Enter the 1st number: 12
Enter the 2nd number: 18
lcm(12,18): 36
```

10. Write a program that prints a diamond shape like the example output. This program reads an odd number from 1 to 19 to determine the number of rows in the diamond.

Assignment #1, Object Oriented Programming, Spring Semester, 2024

Due date: 2024.03.29

```
Enter the number of rows: 3
*
* *
*
```

```
Enter the number of rows: 5
*
* *
*   *
* *
*
```

11. Write a function named *multiple* that takes a pair of integers as arguments. This function should determine if the second integer is a multiple of the first integer, returning true if so, and false otherwise.

```
Enter the 1st number: 3
Enter the 2nd number: 12
multiple(3,12): true
```

```
Enter the 1st number: 5
Enter the 2nd number: 12
multiple(5,12): false
```

12. The Fibonacci series 0, 1, 1, 2, 3, 5, 8, 13, 21, ... starts with the numbers 0 and 1, where each subsequent number is the sum of the two preceding numbers.

(a) Write a non-recursive function *Fibonacci_iter(n)* that calculates the n-th Fibonacci number and return it.

(b) Write a recursive function *Fibonacci_rec(n)* that calculates the n-th Fibonacci number and return it.

```
Enter the number: 9
Fibonacci_iter(9): 21
Fibonacci_rec(9): 21
```

Assignment #1, Object Oriented Programming, Spring Semester, 2024

Due date: 2024.03.29

13. Write a program that converts the lowercase (or uppercase) letter to the uppercase (or lowercase) without using standard library functions in C and C++. (The input string can consist of up to 100 characters.)

```
Enter the string: 2024 KWangWoON uNiveRSiTY.  
Result: 2024 kwANGwOon UnIVersIty.
```

14. An integer is said to be a perfect number if the sum of its factors, including 1 (but not the number itself), is equal to the number. For example, 6 is a perfect number, because $6 = 1 + 2 + 3$.

(a) Write a function *Perfect(n)* that determines whether parameter *n* is a perfect number.

(b) Write a program that prints all the perfect numbers between 1 and 1000. Print the factors of each perfect number to confirm that the number is indeed perfect. You should use the function *Perfect(n)* that you implemented in (a).

```
Perfect numbers between 1 and 1000:  
6 = 1 + 2 + 3  
28 = 1 + 2 + 4 + 7 + 14  
...
```

15. Write the following program. 'Hit & Blow' game is similar to 'Number Baseball' game. When the game start, the program sets four random numbers among 0 to 9, not overlapping. These numbers are not displayed. Whenever you guess and input numbers, you get hints. If there are the same numbers in the same position between random numbers and yours, the program displays "Hit". And if in wrong position, the program displays "Blow". Within given 5 chances, if you obtain correct numbers, this program prints "Win". If you lose, this program prints "Lose" with the correct answer. (The correct answer should be different each time you run it.)

```
Guess: 1234  
  
Hit: 1, Blow: 1  
  
-----  
  
Guess: 3456
```

Assignment #1, Object Oriented Programming, Spring Semester, 2024

Due date: 2024.03.29

Hit: 0, Blow: 3

Guess: 5678

Hit: 2, Blow: 0

Guess: 5632

Hit: 4, Blow: 0

Win

Guess: 1234

Hit: 0, Blow: 0

Guess: 3456

Hit: 1, Blow: 0

Guess: 5678

Hit: 1, Blow: 2

Guess: 5690

Hit: 0, Blow: 2

Guess: 5612

Hit: 0, Blow: 1

Assignment #1, Object Oriented Programming, Spring Semester, 2024

Due date: 2024.03.29

Lose

the correct answer: 9876