

# 객체지향프로그래밍

2-1

아래 모든 과제는 Array를 사용하지 않고 동적할당을 이용해 작성한다.

메모리 누수가 있을 시 감점

1. 10개의 정수(0 ~ 9)를 무작위(rand함수 이용)로 각각 다른 변수에 저장하고 저장된 변수의 값과 메모리 그리고 최대, 최소값을 저장한 변수의 메모리와 값을 출력하는 프로그램을 작성하십시오.

```
Memory Address is 0000003C82D4F798
Value is 1
Memory Address is 0000003C82D4F79C
Value is 7
Memory Address is 0000003C82D4F7A0
Value is 4
Memory Address is 0000003C82D4F7A4
Value is 0
Memory Address is 0000003C82D4F7A8
Value is 9
Memory Address is 0000003C82D4F7AC
Value is 4
Memory Address is 0000003C82D4F7B0
Value is 8
Memory Address is 0000003C82D4F7B4
Value is 8
Memory Address is 0000003C82D4F7B8
Value is 2
Memory Address is 0000003C82D4F7BC
Value is 4
Max Data is 9(0000003C82D4F7A8)
Min Data is 0(0000003C82D4F7A4)
```

<출력 예시>

2. 2개의 문자열을 입력 받아 두 문자열을 이어 붙인 새로운 문자열을 출력하는 프로그램을 작성하십시오. 출력할 때 두개의 문자열을 이어 출력하지 않고 새롭게 두 개의 문자열을 이어붙인 문자열을 만들어 출력하며 필요한 모든 함수는 직접 구현해 사용한다. (STL, strcpy, strcat 등 사용불가)

```
String 1 :Objected Oriented
String 2 : Programing
```

<입력 예시>

```
String 1 :Objected Oriented
String 2 : Programing
Result : Objected Oriented Programing
```

<출력 예시>

3. 2개의 Matrix의 크기(row, column)를 입력 받아 내부를 10이하의 무작위 자연수로 채운 입력 받은 크기의 Matrix들을 출력하고 만약 두 Matrix가 Matrix Multiplication을 할 수 있다면 두 Matrix의 곱셈 결과를 출력하는 프로그램을 작성하시오. 이 때 필요한 모든 함수는 직접 구현 해서 사용한다.

```
Matrix A : 4 3
Matrix B : 3 4

A Matrix :
1 7 4
0 9 4
8 8 2
4 5 5

B Matrix :
1 7 1 1
5 2 7 6
1 4 2 3

A*B Result :
40 37 58 55
49 34 71 66
50 80 68 62
34 58 49 49
```

<예시 1>

```
Matrix A : 3 3
Matrix B : 4 3

A Matrix :
1 7 4
0 9 4
8 8 2

B Matrix :
4 5 5
1 7 1
1 5 2
7 6 1

Can't Operate Matrix Multiplication(4!=3)
```

<예시 2>

4. 0 ~ 100까지 임의의 자연수로 채워져 있는 Matrix를 int\*\*형 변수에 저장해 출력하고 Matrix를 행 단위로 오름차순 정렬한 후 재 출력, 행의 총 합을 기준으로 오름차순으로 정렬한 후 재 출력하는 프로그램을 작성하시오. 이 때 행의 총 합을 기준으로 정렬할 때 값을 직접 바꾸지 않고 포인터가 가리키는 주소를 바꿔 정렬한다.

```
Original Matrix
41 67 34 0 69 24 78 58 62 64
5 45 81 27 61 91 95 42 27 36
91 4 2 53 92 62 21 16 18 95
47 26 71 38 69 12 67 99 35 94
9 11 22 39 79 64 41 11 53 68
47 44 62 57 37 59 23 41 29 78
16 35 90 42 88 6 40 42 64 48
46 5 90 29 70 50 6 1 93 48
29 23 84 54 56 40 66 76 31 8
44 39 26 23 37 38 18 82 29 41

Sort by row
0 24 34 41 58 62 64 67 69 78 Sum is 497(00000232BB70E730)
5 27 27 36 42 45 61 81 91 95 Sum is 510(00000232BB70E650)
91 4 16 18 21 53 62 91 92 95 Sum is 474(00000232BB70E500)
12 26 35 38 47 67 69 71 94 99 Sum is 558(00000232BB70E1F0)
9 11 11 22 33 41 53 64 68 73 Sum is 379(00000232BB70E400)
23 29 37 41 44 47 57 59 82 78 Sum is 477(00000232BB70E1B0)
6 16 35 40 42 42 48 64 88 90 Sum is 471(00000232BB70E5E0)
1 5 6 29 46 48 50 70 90 93 Sum is 498(00000232BB70E6C0)
8 23 29 31 40 54 56 66 76 84 Sum is 467(00000232BB70E7A0)
18 23 26 29 37 38 39 41 44 82 Sum is 377(00000232BB70E810)

Sort by Sum
18 23 26 29 37 38 39 41 44 82 Sum is 377(00000232BB70E810)
9 11 11 22 33 41 53 64 68 73 Sum is 379(00000232BB70E420)
1 5 6 29 46 48 50 70 90 93 Sum is 498(00000232BB70E6C0)
8 23 29 31 40 54 56 66 76 84 Sum is 467(00000232BB70E7A0)
6 16 35 40 42 42 48 64 88 90 Sum is 471(00000232BB70E5E0)
2 4 16 18 21 53 62 91 92 95 Sum is 474(00000232BB70E500)
23 29 37 41 44 47 57 59 82 78 Sum is 477(00000232BB70E1B0)
0 24 34 41 58 62 64 67 69 78 Sum is 497(00000232BB70E730)
5 27 27 36 42 45 61 81 91 95 Sum is 510(00000232BB70E650)
12 26 35 38 47 67 69 71 94 99 Sum is 558(00000232BB70E1F0)
```

<예시>