

객체지향프로그래밍실습 2-1

학부: 컴퓨터정보공학부

학번: 2019202021

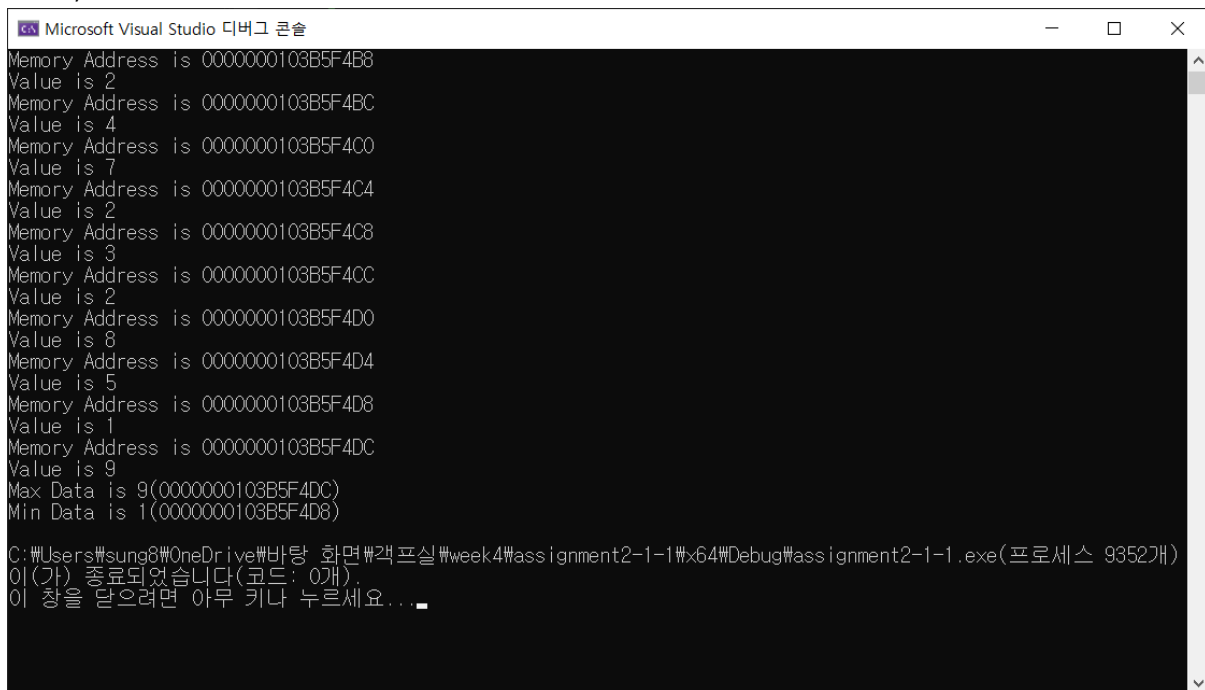
이름: 정성엽

<1번 문제>

1) 문제설명

- rand함수를 이용하여 0~10까지 임의의 수를 다른 변수에 저장하되 srand(time(NULL)); 코드를 사용하여 실행할 때마다 다른 값이 나올 수 있도록 한다. 변수에 저장된 값과 주소 값이 출력되도록 하며 후에 최대와 최소값을 저장한 변수의 값과 주소 값이 출력되도록 한다.

2) 결과화면



```
Microsoft Visual Studio 디버그 콘솔
Memory Address is 0000000103B5F4B8
Value is 2
Memory Address is 0000000103B5F4BC
Value is 4
Memory Address is 0000000103B5F4C0
Value is 7
Memory Address is 0000000103B5F4C4
Value is 2
Memory Address is 0000000103B5F4C8
Value is 3
Memory Address is 0000000103B5F4CC
Value is 2
Memory Address is 0000000103B5F4D0
Value is 8
Memory Address is 0000000103B5F4D4
Value is 5
Memory Address is 0000000103B5F4D8
Value is 1
Memory Address is 0000000103B5F4DC
Value is 9
Max Data is 9(0000000103B5F4DC)
Min Data is 1(0000000103B5F4D8)

C:\Users\#sung8\OneDrive\바탕 화면\객체프실\week4\assignment2-1-1\#x64\Debug\assignment2-1-1.exe(프로세스 9352개)
이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요....
```

- 임의의 수 저장 및 출력, 최대값 최소값 각각 값과 주소 출력한 결과화면

3) 고찰

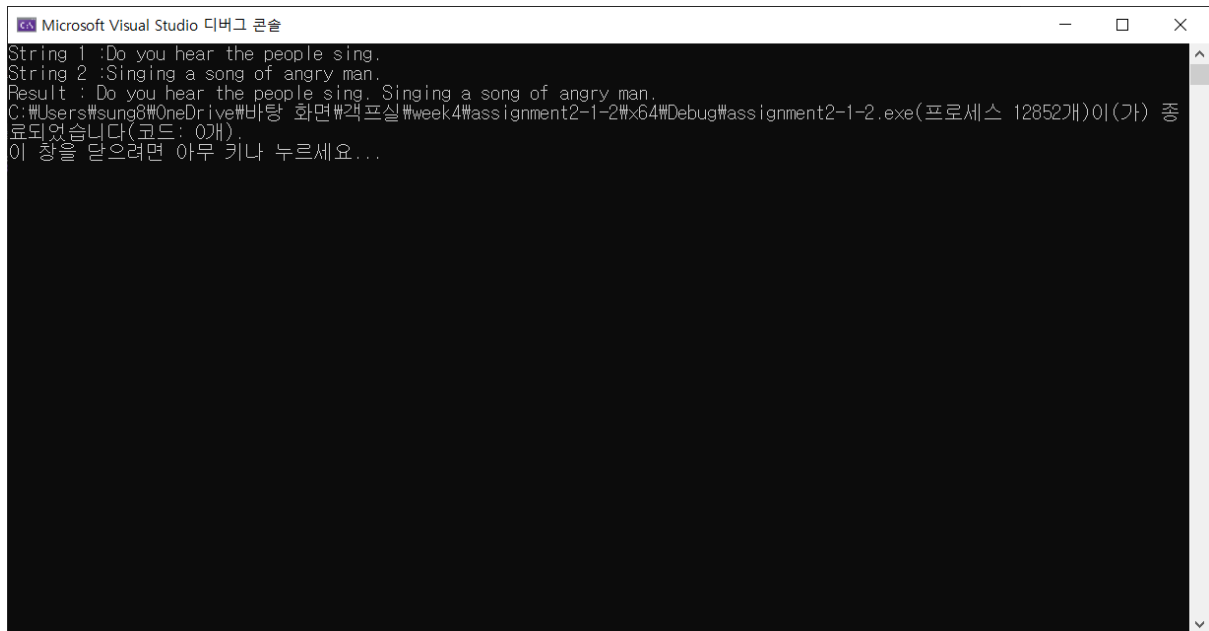
- 최대와 최소값을 저장하는 것은 일반 변수로도 가능하지만 해당 주소를 가져올 수 없다. 그러므로 포인터변수를 사용하여 해당 값은 물론 주소를 가리키도록 하여 값과 주소 값 모두 가져와 출력하도록 하였다.

<2번 문제>

1) 문제설명

- 2개의 문자열을 입력 받고 이를 합친 문자열을 출력하는 프로그램으로 getline함수를 통해 해당 문자열을 입력 받고 후에 NULL값이 있는 곳을 찾아 strlen 함수 대신 문자열 길이를 알아내도록 한다. 이 알아낸 길이들을 통해 새로운 포인터 변수를 선언하고, (길이의 합 +2) 만큼 동적할당을 해야 한다. 이 후에는 함수를 만들어 이 두 문자열을 합친 것을 새로운 변수에 저장한다.

2) 결과화면



```
Microsoft Visual Studio 디버그 콘솔
String 1 :Do you hear the people sing.
String 2 :Singing a song of angry man.
Result : Do you hear the people sing. Singing a song of angry man.
C:\Users\sung8\OneDrive\바탕 화면\백프실\week4\assignment2-1-2\x64\Debug\assignment2-1-2.exe(프로세스 12852개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

- 문자열 2개를 입력했을 때 결과화면

3) 고찰

- 두 문자열을 합칠 때 그 중간에는 띄어쓰기가 들어가고 마지막 문장에 널문자를 넣기 위해 각 길이의 합에 2를 더한만큼 동적할당을 했으며 후에 메모리 누수를 없애기 위해 delete 명령어를 이용했다. 문장을 합칠 때는 함수를 만들어 함수 내에서 이 두 문장이 합쳐지지만 각 매개변수는 각 함수의 주소를 나타내기 때문에 합쳐진 문자열이 main에서도 출력할 수 있다.

<3번 문제>

1) 문제설명

- 2개의 Matrix가 각각 row와 column 값을 입력 받고 해당 만큼 등적할당을 하며 각각에 10 이하의 무작위 수를 저장한다. 이후에 이 Matrix가 Matrix Multiplication을 할 수 있을 경우에는 곱셈 결과를 출력하며 불가능하다면 불가능함을 출력하도록 한다.

2) 결과화면

```
선택 Microsoft Visual Studio 디버그 콘솔
Matrix A :3 4
Matrix B :4 3

A Matrix :
2      0      4      0
6      8      4      5
9      7      3      6

B Matrix :
10     5      4
8      8      5
4      0      2
7      10     5

A*B Result :
36     10     16
175    144    97
200    161    107

C:\Users\sung8\OneDrive\바탕 화면\객포실\week4\assignment2-1-3\Debug\assignment2-1-3.exe(프로세스 4992개)이(가) 종료
되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

- 3 4 4 3 을 입력했을 때 결과화면

```
Microsoft Visual Studio 디버그 콘솔
Matrix A :3 5
Matrix B :4 3

A Matrix :
6      1      3      9      9
1      6      9      3      7
2      8      9      0      3

B Matrix :
5      0      7
10     2      4
7      0      5
5      0      6

Can't Operate Matrix Multiplication(4!=5)

C:\Users\sung8\OneDrive\바탕 화면\객포실\week4\assignment2-1-3\Debug\assignment2-1-3.exe(프로세스 8484개)이(가) 종료
되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

- 3 5 4 3을 입력했을 때 결과화면

3) 고찰

- 먼저 문제에서는 10이하의 무작위 자연수라고 표현하지만 문제와 함께 주어진 예시에는 '0'이 포함되었기에 따로 0을 제외하고 저장하지 않았다. Matrix Multiplication이 무엇인지와 방법 알고 이것이 되는 경우와 안 되는 경우를 알아야하는 문제로, Matrix A의 column과 Matrix B의 row가 같은 경우에 Matrix Multiplication이 가능하며 이때 곱해진 결과의 Matrix의 row는 Matrix A의 row이고 column은 Matrix B의 column임을 유의하여 함수를 만들었다.

<4번 문제>

1) 문제 설명

- 10x10 matrix에 0~100까지의 무작위 수를 저장하고 int**형 변수에 이를 저장해 출력하도록 한다. 처음엔 행 단위로 오름차순 정렬한 후 재 출력하며, 두 번째엔 행의 총 합을 기준으로 오름차순으로 정렬한다. 이때 두 번째 같은 경우 값을 직접 바꾸는 것이 아닌 포인터가 가리키는 주소를 바꿔 정렬한다.

2) 결과화면

```
Microsoft Visual Studio 디버그 콘솔
Original Matrix
82 48 20 60 95 77 51 44 18 41
68 31 74 94 88 46 41 95 89 22
59 59 53 77 55 12 92 21 82 97
53 61 98 24 4 8 97 100 58 77
21 62 84 32 27 94 11 52 31 4
29 47 84 73 49 40 37 54 13 47
21 5 71 54 71 45 50 45 30 16
13 23 80 77 57 41 0 24 49 41
70 94 81 58 92 22 13 45 91 76
29 77 31 87 78 13 10 25 90 94

Sort by row
18 20 41 44 48 51 60 77 82 95 Sum is 536(000000D9DCCFF648)
22 31 41 46 68 74 88 89 94 95 Sum is 648(000000D9DCCFF64C)
12 21 53 55 59 59 77 82 92 97 Sum is 607(000000D9DCCFF650)
4 8 24 53 58 61 77 97 98 100 Sum is 580(000000D9DCCFF654)
4 11 21 27 31 32 52 62 84 94 Sum is 418(000000D9DCCFF658)
13 29 37 40 47 47 49 54 73 84 Sum is 473(000000D9DCCFF65C)
5 16 21 30 45 45 50 54 71 71 Sum is 408(000000D9DCCFF660)
0 13 23 24 41 41 49 57 77 80 Sum is 405(000000D9DCCFF664)
13 22 45 58 70 76 81 91 92 94 Sum is 642(000000D9DCCFF668)
10 13 25 29 31 77 78 87 90 94 Sum is 534(000000D9DCCFF66C)

Sort by Sum
22 31 41 46 68 74 88 89 94 95 Sum is 648(000000D9DCCFF64C)
13 22 45 58 70 76 81 91 92 94 Sum is 642(000000D9DCCFF668)
12 21 53 55 59 59 77 82 92 97 Sum is 607(000000D9DCCFF650)
4 8 24 53 58 61 77 97 98 100 Sum is 580(000000D9DCCFF654)
22 31 41 46 68 74 88 89 94 95 Sum is 536(000000D9DCCFF648)
10 13 25 29 31 77 78 87 90 94 Sum is 534(000000D9DCCFF66C)
10 13 25 29 31 77 78 87 90 94 Sum is 473(000000D9DCCFF65C)
22 31 41 46 68 74 88 89 94 95 Sum is 418(000000D9DCCFF658)
10 13 25 29 31 77 78 87 90 94 Sum is 408(000000D9DCCFF660)
22 31 41 46 68 74 88 89 94 95 Sum is 405(000000D9DCCFF664)

C:\Users\#sung8\OneDrive\바탕 화면\2객 프실\week4\assignment2-1-4\#Debug\assignment2-1-4.exe(프로세스 11660개)이(가) 종료
되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

- 임의의 수가 저장되고 행 단위, 합을 기준으로 오름차순으로 정렬한 결과화면

3) 고찰

- 임의의 int**형 변수에 무작위 값을 저장하고 Original Matrix를 표현할 이중 포인터 변수를 선언하고 이것이 임의의 변수를 가리키도록 프로그래밍했다. 처음 행 단위로 오름차순으로 정렬할 경우 단순히 tmp라는 임의의 변수를 선언하여 위치를 바꿨다. 하지만 다음 행의 총 합을 기준으로 정렬할 때는 먼저 합이 가장 큰 순이 몇 번째인지 저장하고 해당 순서대로 Original Matrix의 행이 임의의 변수 행 첫번째 주소와 같음을 이용하여 각 주소를 가리켜 순서를 변경하도록 했다.