객체지향프로그래밍 및 실습

OOP 2-1 과제

학과 : 컴퓨터정보공학부
학번 : 2021202043
이름 : 이은서
제출일 : 2022.04.10

문제 1.

1. 문제 설명

10개의 정수를 무작위로 변수에 저장하고 그 변수의 값과 메모리, 그리고 최대, 최소값을 저장한 변수의 값과 메모리를 출력하는 프로그램을 구현한다.

2. 결과 화면

```
"/Users/ieunseo/Downloads/practice C++/cmake-build-debug/2-1-1"
Memory Address is 0x600003e98210
Value is 1
Memory Address is 0x600003e98214
Value is 9
Memory Address is 0x600003e98218
Value is 3
Memory Address is 0x600003e9821c
Value is 0
Memory Address is 0x600003e98220
Value is 1
Memory Address is 0x600003e98224
Value is 2
Memory Address is 0x600003e98228
Value is 5
Memory Address is 0x600003e9822c
Value is 6
Memory Address is 0x600003e98230
Value is 5
Memory Address is 0x600003e98234
Value is 3
Max Data is 9(0x600003e98214)
Min Data is 0(0x600003e9821c)

종료 코드 0(으)로 완료된 프로세스
```

3. 고찰

Max Data, Min Data에 값을 저장하는 것은 문제없었지만, 메모리의 값을 같이 저장하는 부분에서 문제가 발생했다. 따라서 주소값을 저장하는 변수를 하나씩 더 만들어서 출력하도록 하였고 문제를 해결하였다.

문제 2.

1. 문제 설명

2개의 문자열을 입력 받아 두 문자열을 이어 붙인 새로운 문자열을 출력

2. **결과 화면**

```
"/Users/ieunseo/Downloads/practice C++/cmake-build-debug/2-1-2"
String 1: Objected Oriented
String 2:  Programing
Reuslt: Objected Oriented Programing

종료 코드 0(으)로 완료된 프로세스
```

3. **고찰**

문자열이 길어지면 뒤에 문자를 먹거나 두번씩 나오는 등의 오류가 발생하였는데, 각각의 길이와 시작점, 끝나는 지점을 정확하게 계산하여 시행착오를 겪은 결과 해결하였다.

문제 3.

1. **문제 설명**

2개의 matrix의 크기를 입력하면, 그 matrix에 10이하의 무작위 자연수로 채운 후 출력하고, 두 matrix를 연산하여 결과를 출력하는 프로그램 구현

2. **결과 화면**

```
"/Users/ieunseo/Downloads/prac
Matrix A :4 3
Matrix B :3 4
A Matrix :
1       2       8
7       5       7
9       1       0
8       3       4

B Matrix :
8       1       1       8
8       9       3       9
6       0       6       9

A*B Result :
72      19      55      98
138     52      64      164
80      18      12      81
112     35      41      127

종료 코드 0(으)로 완료된 프로세스
```

```
"/Users/ieunseo/Downloads/practice C++/cmake
Matrix A :4 3
Matrix B :2 4
A Matrix :
6       8       5
9       2       2
9       8       2
6       0       7

B Matrix :
3       7       1       0
5       4       3       6


Can't Operate Matrix Multiplication(2!=3)
```

3. **고찰**

matrixA, matrixB, matrixAB를 각각 동적할당하고 해제를 하였으나 pointer being freed was not allocated라는 오류가 떴다. 그래서 matrixAB는 해제하지 않은 결과 제대로 값이 나왔다. matrixA, matrixB와 matrixAB가 포함되어있어 앞의 2개의 행렬이 동적 해제되면서 matrixAB에게 할당된 값이 없어진 것이라고 판단했다.

문제 4.

1. 문제 설명

matrix에 0~100까지의 수를 임의로 저장하고 행 단위로 오름차순 정렬 후, 행의 총 합을 기준으로 오름차순으로 정렬하여 재출력한다. 총 합을 기준으로 정렬할 때는 값을 직접 바꾸지 않고, 포인터가 가리키는 주소를 바꿔 정렬한다.

2. 결과 화면

```
"/Users/ieunseo/Downloads/practice C++/cmake-build-debug/2-1-4"
Original Matrix
22      82      6       25      12      89      30      35      2       56
61      89      0       65      43      42      65      33      63      88
14      22      32      73      33      85      9       51      79      43
14      10      31      9       27      35      67      72      62      73
72      22      22      12      14      93      16      28      84      39
48      91      38      36      20      66      93      18      8       83
67      95      4       48      36      64      88      45      40      33
16      71      93      68      94      18      51      78      99      15
23      79      6       81      12      66      28      52      32      91
55      10      53      4       0       27      46      40      44      67

Sort by row
2       6       12      22      25      30      35      56      82      89      Sum is 359(0x600000870238)
0       33      42      43      61      63      65      65      88      89      Sum is 549(0x600000870268)
9       14      22      32      33      43      51      73      79      85      Sum is 441(0x600000870298)
9       10      14      27      31      35      62      67      72      73      Sum is 400(0x6000008702c8)
12      14      16      22      22      28      39      72      84      93      Sum is 402(0x6000008702f8)
8       18      20      36      38      48      66      83      91      93      Sum is 501(0x600000870328)
4       33      36      40      45      48      64      67      88      95      Sum is 520(0x600000870358)
15      16      18      51      68      71      78      93      94      99      Sum is 603(0x600000870388)
6       12      23      28      32      52      66      79      81      91      Sum is 470(0x6000008703b8)
0       4       10      27      40      44      46      53      55      67      Sum is 346(0x6000008703e8)
Sort by sum
0       4       10      27      40      44      46      53      55      67      Sum is 346(0x6000008703e8)
2       6       12      22      25      30      35      56      82      89      Sum is 359(0x600000870238)
9       10      14      27      31      35      62      67      72      73      Sum is 400(0x6000008702c8)
12      14      16      22      22      28      39      72      84      93      Sum is 402(0x6000008702f8)
9       14      22      32      33      43      51      73      79      85      Sum is 441(0x600000870298)
6       12      23      28      32      52      66      79      81      91      Sum is 470(0x6000008703b8)
8       18      20      36      38      48      66      83      91      93      Sum is 501(0x600000870328)
4       33      36      40      45      48      64      67      88      95      Sum is 520(0x600000870358)
0       33      42      43      61      63      65      65      88      89      Sum is 549(0x600000870268)
15      16      18      51      68      71      78      93      94      99      Sum is 603(0x600000870388)

종료 코드 0(으)로 완료된 프로세스
```

3. 고찰

행의 총 합을 기준으로 정렬하는 과정에서 포인터 주소를 서로 바꾸는 과정에서 오류가 자주 발생하였지만 새로운 int * temp를 선언하고, 각각 행의 0번째 column 포인터 주소를 가리키게 하였더니 모든 행이 바뀌었고 문제를 해결할 수 있었다.