



컴퓨터프로그래밍 및 실습

화 3,4 (5313) / 목 1,2 (5402)

RTDCS. 422.

조용아

cyr1212@naver.com



Real-time Operating system for Dependable Embedded Operation
<http://www.rodeolab.org>

컴퓨터 개론 및 실습

➤ Teaching Assistant

- 조용아 (공대 422호)
- E-mail : cyr1212@naver.com

배열(Array)

- 배열(Array) : 같은 종류의 데이터들이 순차적으로 저장되어 있는 자료 구조

배열(Array)

- 개별 변수 사용

```
int S0, S1, S2, S3, S4, S5, S6, S7, S8, S9;
```

- 배열 사용

```
int S[10];
```

첨자 또는 인덱스(인덱스 : 배열 원소의 번호)

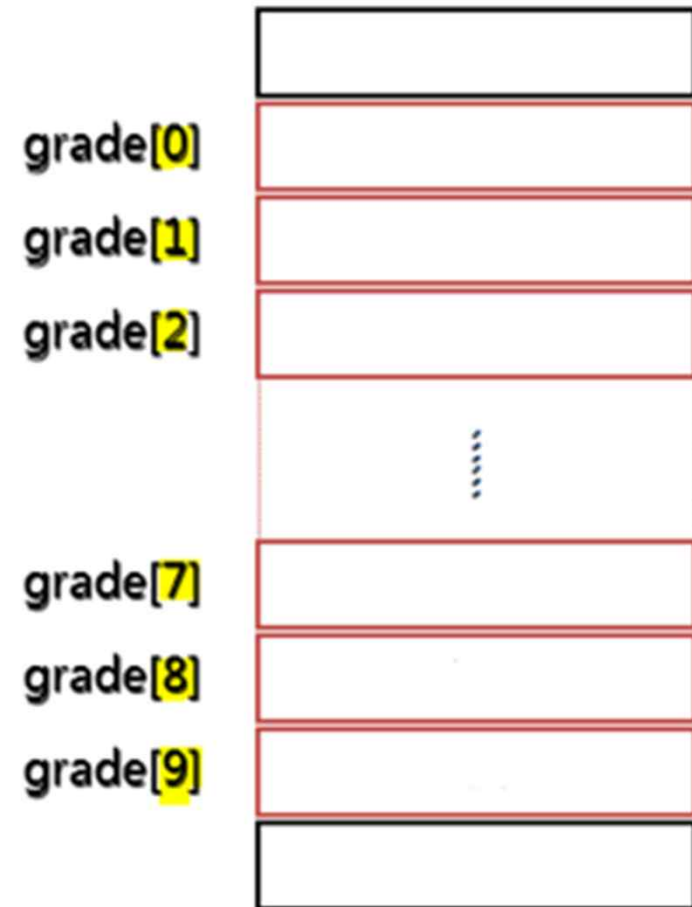
배열(Array) 선언

- 자료형 배열이름[배열 크기];
- 배열 크기 : 항상 상수를 사용
음수 X (ex. ~~int grade[-2];~~)
실수 X (ex. ~~int grade[6.7];~~)
- 배열은 메모리의 연속적인 공간에 저장

배열(Array)의 선언

- 자료형 배열이름[배열크기];
- `int grade[10];`
- 배열 번호는 항상 0부터 시작!

```
int score[60];  
float cost[12];  
char name[50];  
char src[10], dst[10];  
int index, days[7];
```



배열(Array)의 접근

- int grade[10];

grade[0] = 87;
grade[1] = 74;
grade[2] = 80;
⋮
grade[7] = 67;
grade[8] = 97;
grade[9] = -20;

grade[0]	87
grade[1]	74
grade[2]	80
	⋮
grade[7]	67
grade[8]	94
grade[9]	-20

배열(Array)의 초기화

- `int grade[10] = {87, 74, 80, 10, 20, 30, 40, 67, 94, -20};`
- `int grade[10] = {87, 74, 80};`

grade[0]	87
grade[1]	74
grade[2]	80
	⋮
grade[7]	67
grade[8]	94
grade[9]	-20

grade[0]	87
grade[1]	74
grade[2]	80
	⋮
grade[7]	0
grade[8]	0
grade[9]	0

배열(Array)의 초기화

- int grade[] = {87, 74, 80, 10, 20, 30, 40, 67, 94, -20};

grade[0]	87
grade[1]	74
grade[2]	80
	⋮
grade[7]	67
grade[8]	94
grade[9]	-20

배열(array) 예제 1

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int grade[5];
8      int i;
9
10     grade[0] = 10;
11     grade[1] = 20;
12     grade[2] = 30;
13     grade[3] = 40;
14     grade[4] = 50;
15
16     for(i=0; i<5; i++)
17         cout << "grade[" << i << "] = " << grade[i] << endl;
18
19     return 0;
20 }
```

배열(array) 예제 2

```
1  #include <iostream>
2  #define SIZE 5
3
4  using namespace std;
5
6  int main()
7  {
8      int grade[SIZE];
9      int i;
10
11     cout << "5명의 점수를 입력하세요 : ";
12
13     for(i=0; i<SIZE; i++)
14         cin >> grade[i];
15
16     for(i=0; i<5; i++)
17         cout << "grade[" << i << "] = " << grade[i] << endl;
18
19     return 0;
20 }
```

배열(array)의 복사

```
int a[5] = {1, 2, 3, 4, 5};  
int b[5];  
  
b = a;
```

배열의 이름을 다른 배열에
대입한다고 해서 배열이 복사
되지 않는다.
배열의 이름은 배열의 시작을
가르킨다.

```
int i;  
int a[5] = {1, 2, 3, 4, 5};  
int b[5];  
  
for(i=0; i<5; i++)  
    b[i] = a[i];
```

하나의 배열을 다른 배열로
복사하기 위해서는 각각의 원
소를 복사해주어야 한다.

배열(array)의 비교

```
int a[5] = {1, 2, 3, 4, 5};  
int b[5] = {1, 2, 3, 4, 5};
```

```
if(a == b)  
    printf("같습니다.");  
else  
    printf("다릅니다.");
```

```
int i;  
int a[5] = {1, 2, 3, 4, 5};  
int b[5] = {1, 2, 3, 4, 5};
```

```
for(i=0; i<5; i++)  
{  
    if(a[i] != b[i])  
    {  
        printf("다릅니다.");  
        break;  
    }  
}
```

2차원 배열

- `int s[10];` // 1차원 배열

`int s[2][3];` // 2차원 배열

`int s[2][3][4];` // 3차원 배열

2차원 배열

- int s[2][3];
 행 열

s[0][0]	s[0][1]	s[0][2]
s[1][0]	s[1][1]	s[1][2]

행

s[0][0]	s[0][1]	s[0][2]
s[1][0]	s[1][1]	s[1][2]

열

2차원 배열의 초기화

- `int s[2][3] = { {0, 1, 2}, {3, 4, 5} };`

0	1	2
3	4	5

- `int s[][3] = { {0, 1, 2}, {3, 4, 5} };`

0	1	2
3	4	5

- `int s[][3] = { {0, 1}, {3, 4} };`

0	1	0
3	4	0

- `int s[][3] = {0, 1, 2, 3, 4, 5};`

0	1	2
3	4	5

2차원 배열 예제

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int arr[2][3] = {{0, 1, 2}, {3, 4, 5}};
8      int i, j;
9
10     for(i=0; i<2; i++)
11     {
12         for(j=0; j<3; j++)
13         {
14             cout << arr[i][j] << " ";
15         }
16         cout << endl;
17         //cout << "\n";
18     }
19
20     return 0;
21 }
```

0	1	2
3	4	5

0	1	2
3	4	5

3차원 배열 예제 1

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int arr[3][3][3] = {0, };
8      int x, y, z;
9      int i = 1;
10
11     for(x=0; x<3; x++)
12     {
13         for(y=0; y<3; y++)
14         {
15             for(z=0; z<3; z++)
16             {
17                 arr[x][y][z] = i++;
18             }
19         }
20     }
21
22     return 0;
23 }
```

			19	20	21
		10	11	12	24
1	2	3	15	27	
4	5	6	18		
7	8	9			

3차원 배열 예제 2

```
21
22     for(x=0; x<3; x++)
23     {
24         for(y=0; y<3; y++)
25         {
26             for(z=0; z<3; z++)
27             {
28                 cout << arr[x][y][z] << " ";
29             }
30             cout << endl;
31         }
32         cout << endl;
33     }
34
35     return 0;
36 }
```

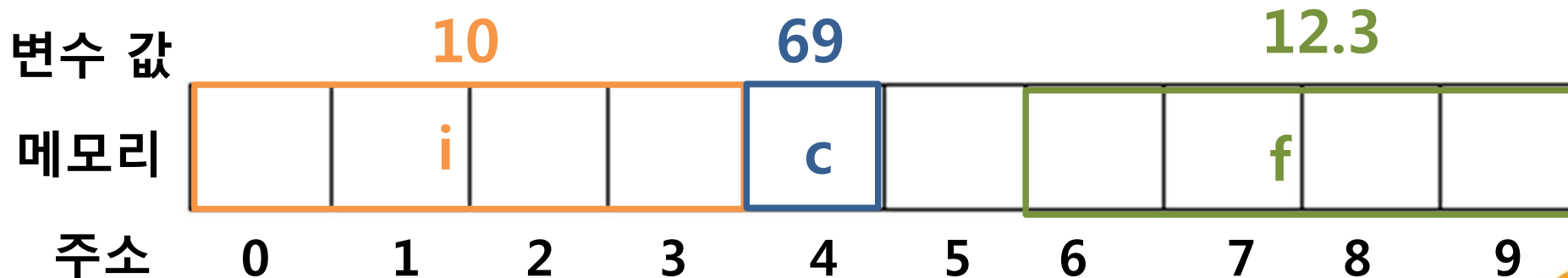
1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
16	17	18
19	20	21
22	23	24
25	26	27

포인터(Pointer)

- 포인터(Pointer) : 메모리의 주소를 가지고 있는 변수

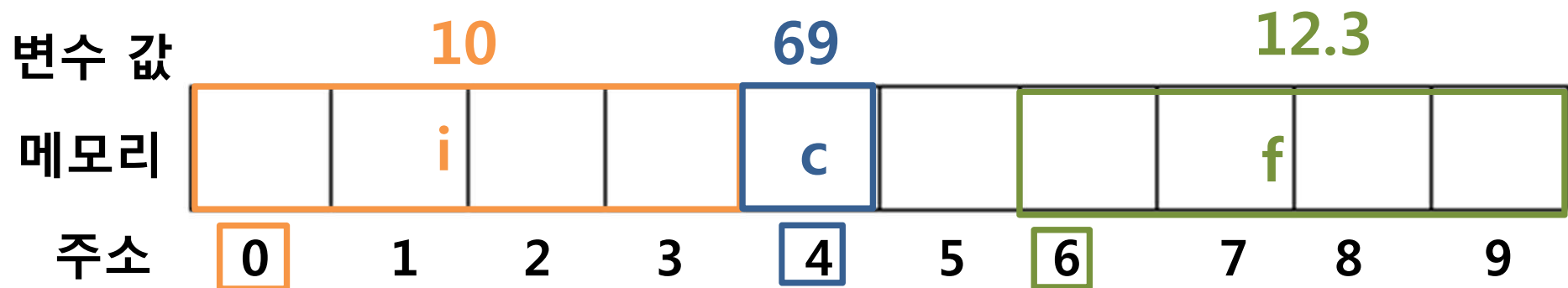
변수의 메모리 저장 1

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int i = 10;
8      char c = 69;
9      float f = 12.3;
10
11      return 0;
12 }
```



변수의 메모리 저장 2

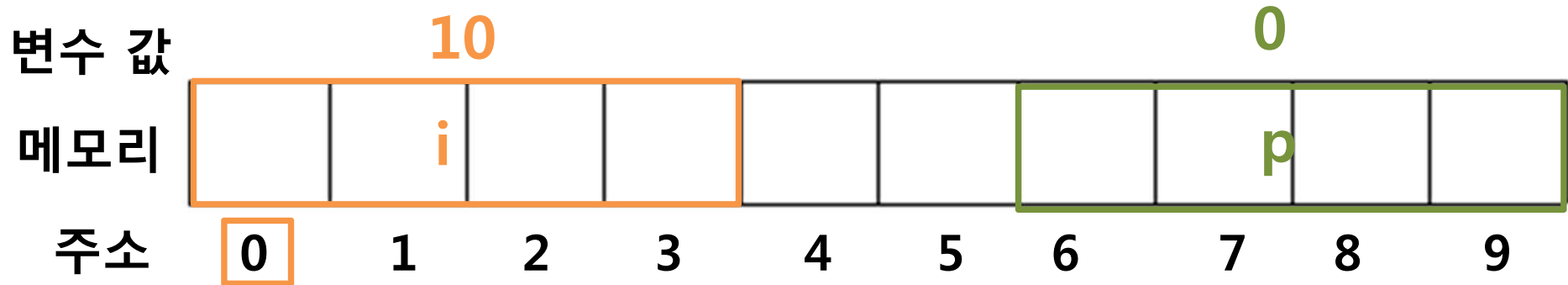
- &(주소 연산자) : 변수의 주소를 계산하는 연산자



```
cout << &i << endl; // -> 0
cout << &c << endl; // -> 4
cout << &f << endl; // -> 6
```

포인터(Pointer)

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int i = 10;
8     int *p = &i;
9
10    cout << *p;
11
12    return 0;
13 }
```



포인터(Pointer)

- *(간접 참조 연산자) : 포인터가 가리키는 값을 가져오는 연산자

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int i = 10;
8      int *p = &i;
9
10     cout << *p;
11
12     return 0;
13 }
```

100이 출력

포인터 연산

- 증가, 감소, 덧셈, 뺄셈의 연산이 가능
- 포인터 타입의 크기만큼 증감

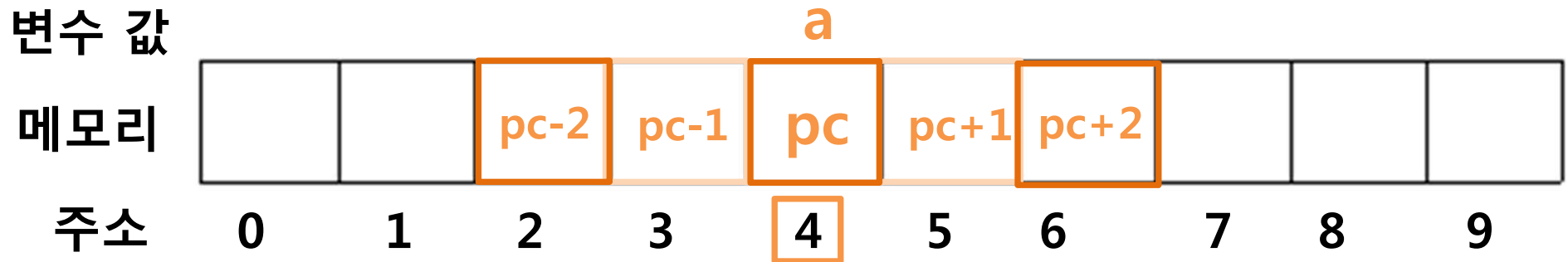
포인터 타입	++연산 후 증가되는 값
char	1
short	2
int	4
float	4
double	8

포인터 연산 예제 1

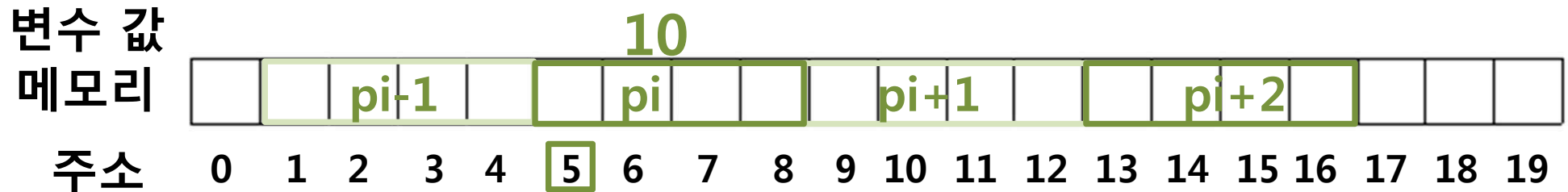
```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      char *pc;
8      int *pi;
9      double *pd;
10
11     pc = (char *)10000;
12     pi = (int *)10000;
13     pd = (double *)10000;
14
15     cout << "증가 전 pc = " << (void *)pc << " pi = " << pi << " pd = " << pd << endl;
16
17     pc++;
18     pi++;
19     pd++;
20
21     cout << "증가 후 pc = " << (void *)pc << " pi = " << pi << " pd = " << pd << endl;
22
23     return 0;
24 }
```

포인터 연산 예제 2

- char형 포인터의 증감



- int형 포인터의 증감



포인터와 배열

- 배열 이름 = 포인터
- 배열 이름은 첫 번째 배열 원소의 주소와 같음

포인터와 배열 예제 1

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int arr[] = {10, 20, 30, 40};
8
9      cout << "arr[0] = " << &arr[0] << endl;
10     cout << "arr[1] = " << &arr[1] << endl;
11     cout << "arr[2] = " << &arr[2] << endl;
12
13     cout << "arr = " << &arr << endl;
14     return 0;
15 }
```

포인터와 배열 예제 2

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      const int STUDENTS = 5;
8      int grade[STUDENTS] = {10, 20, 30, 40, 50};
9
10     for(int *p=grade, *pend=grade+STUDENTS; p!=pend; p++)
11         cout << *p << " ";
12
13     return 0;
14 }
```

배열과 함수

- 배열 이름 = 포인터
- 배열 이름은 첫 번째 배열 원소의 주소와 같음
- 배열을 함수의 인수로 전달하는 경우 배열 이름이 포인터이므로 배열의 첫 번째 배열 원소의 주소가 전달

배열과 함수 예제

```
1 #include <iostream>
2
3 int get_avg(const int score[], int n);
4 void increment(int score[], int n);
5
6 using namespace std;
7
8 int main()
9 {
10     const int STUDENTS = 5;
11     int grade[STUDENTS] = {1, 2, 3, 4, 5};
12     int avg;
13
14     increment(grade, STUDENTS);
15     avg = get_avg(grade, STUDENTS);
16     cout << "평균 " << avg << endl;
17
18     return 0;
19 }
20
```

```
21 void increment(int score[], int n)
22 {
23     int i;
24
25     for(i=0; i<n; i++)
26         ++score[i];
27 }
28
29 int get_avg(const int score[], int n)
30 {
31     int i;
32     int sum = 0;
33
34     for(i=0; i<n; i++)
35         sum += score[i];
36
37     return sum/n;
38 }
```


실습 1

- 포인터 연산 예제 직접 코딩 후 결과 첨부

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     char *pc;
8     int *pi;
9     double *pd;
10
11     pc = (char *)10000;
12     pi = (int *)10000;
13     pd = (double *)10000;
14
15     cout << "증가 전 pc = " << (void *)pc << " pi = " << pi << " pd = " << pd << endl;
16
17     pc++;
18     pi++;
19     pd++;
20
21     cout << "증가 후 pc = " << (void *)pc << " pi = " << pi << " pd = " << pd << endl;
22
23     return 0;
24 }
```

실습 2 (1/2)

- 1차원 배열의 합과 평균 출력 프로그램
- 크기가 10인 배열
- 각 배열의 값을 입력 받음
- 배열의 모든 원소의 합과 평균을 출력

실습 2 (2/2)

1 2 3 4 5 6 7 8 9 10

배열의 합 = 55

배열의 평균 = 5

2 4 6 8 10 12 14 16 18 20

배열의 합 = 110

배열의 평균 = 11

1 3 5 7 9 11 13 15 17 19

배열의 합 = 100

배열의 평균 = 10

실습 3 (1/2)

- 행렬의 회전 프로그램
- [3][3], [5][5], [7][7]을 가지는 각각의 행렬을 만듦
- 행렬의 값은 1부터 순서대로 채워짐
(열->행)
- 90도 회전한 행렬을 출력
(왼쪽으로 회전)

실습 3 (2/2)

1	2	3
4	5	6
7	8	9



3	6	9
2	5	8
1	4	7

3	6	9
2	5	8
1	4	7

실습 4 (1/2)

- a^b 의 계산 결과 및 갯수 출력 프로그램

- a와 b의 범위 : $2 \leq a \leq 5$, $2 \leq b \leq 5$

- $2^2=4$, $2^3=8$, $2^4=16$, $2^5=32$
 $3^2=9$, $3^3=27$, $3^4=81$, $3^5=243$
 $4^2=16$, $4^3=64$, $4^4=256$, $4^5=1024$
 $5^2=25$, $5^3=125$, $5^4=625$, $5^5=3125$

- 중복된 수를 제외하면 15개의 숫자

4, 8, 9, 16, 25, 27, 32, 64, 81, 125, 243, 256, 625, 1024, 3125

실습 4 (2/2)

- 계산 결과를 오름차순으로 정렬
- 다음과 같은 결과 화면 첨부

```
count = 15  
4, 8, 9, 16, 25, 27, 32, 64, 81, 125, 243, 256, 625, 1024, 3125
```