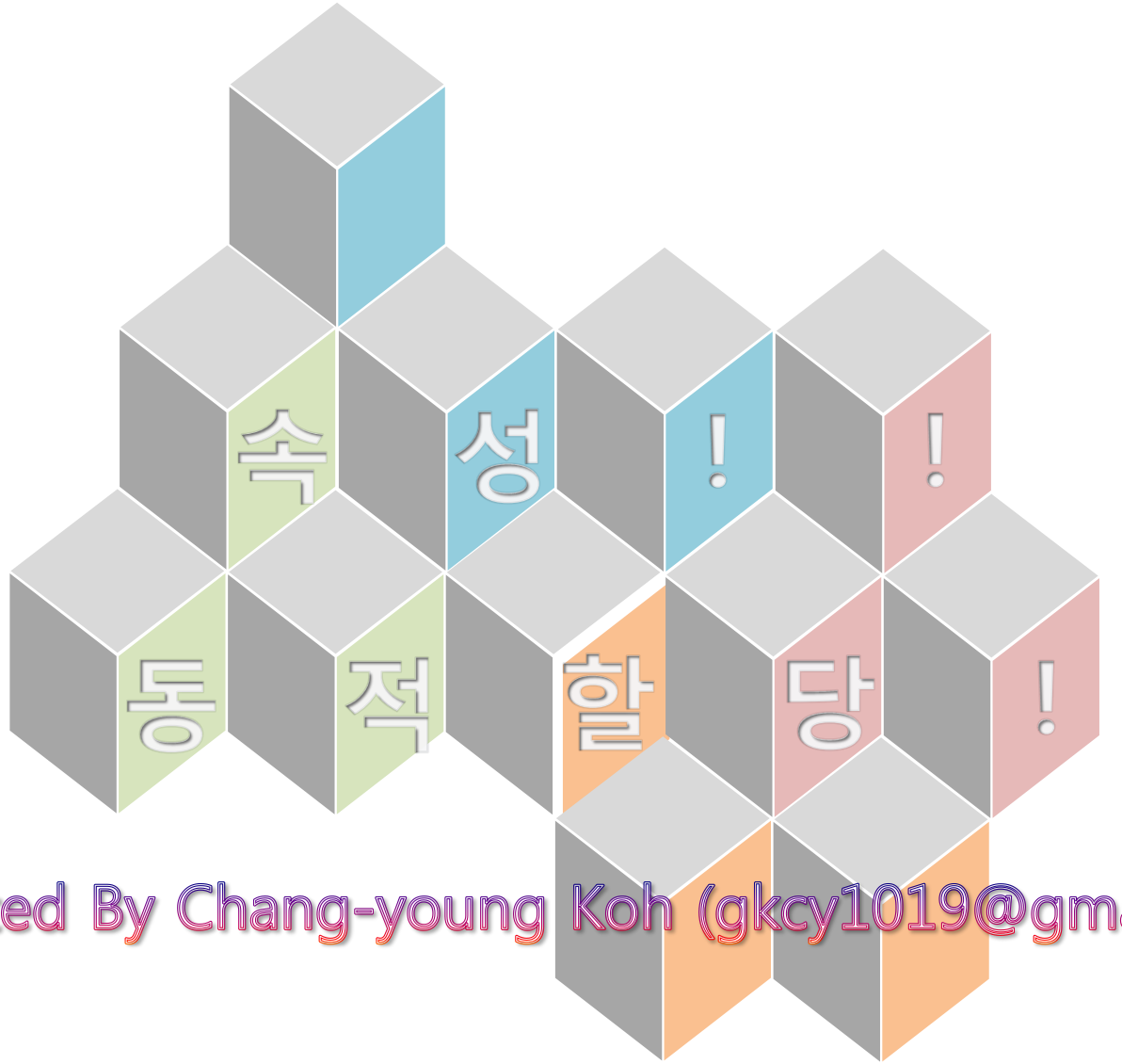
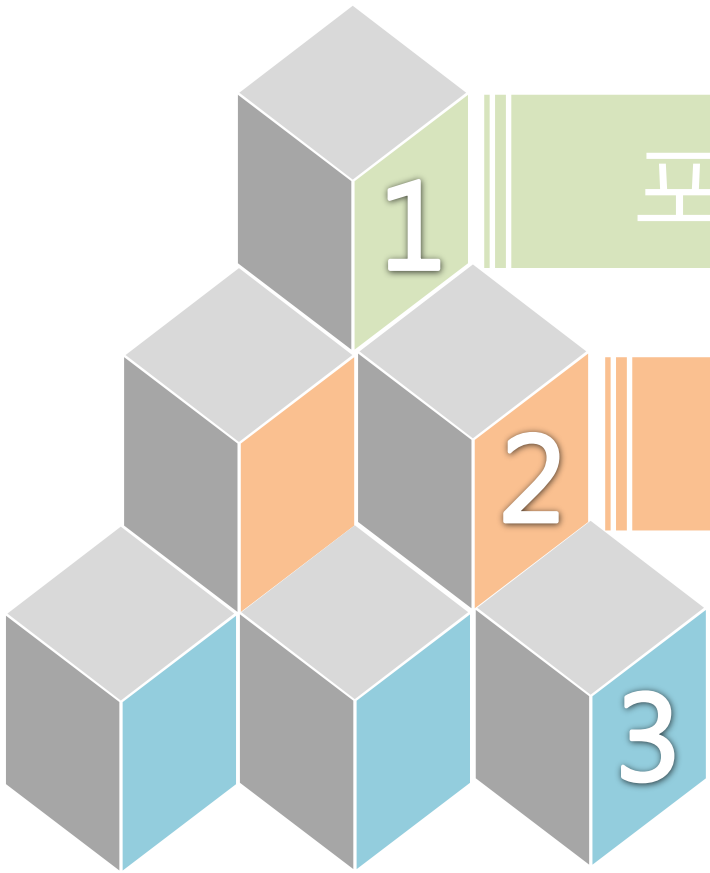
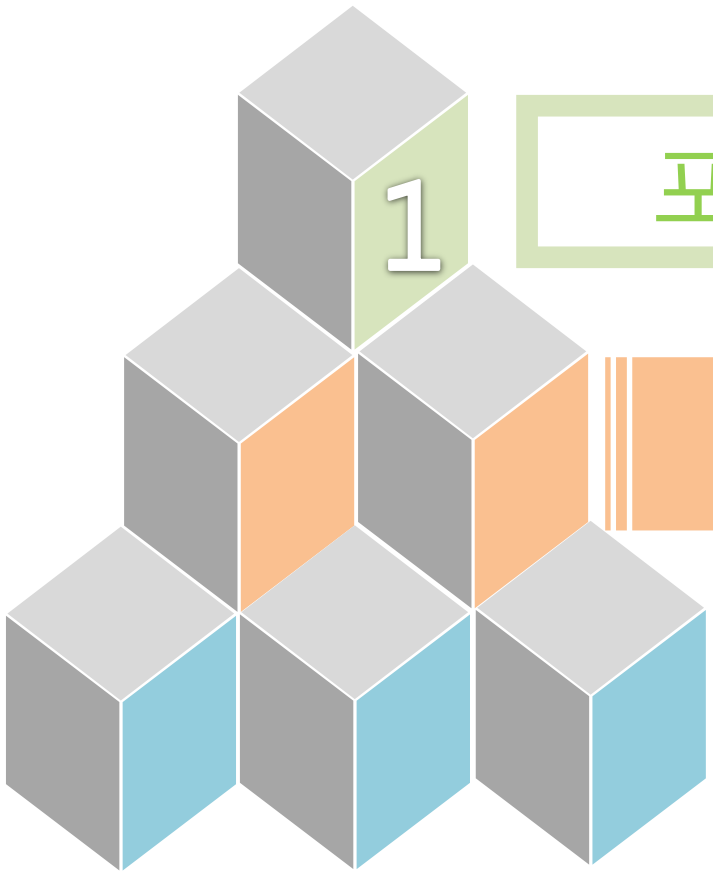


Created By Chang-young Koh (gkcy1019@gmail.com)

무엇을 배우나?



1. 포인터 사용법
2. 동적할당 사용법
3. 포인터와 동적할당



포인터 사용법

포인터 사용법

```
void swap (int *x, int *y)
{
        int tmp = *y;
        *y = *x;
        *x = *y;
}
```

우린 보통 함수 내에서 값을 바꾸는 데에 썼다.
Call By Reference.

알아두고 넘어가야 하는 것 :
int x;
*(&x) 와 x는 같다.
&x : x의 주소.
*(&x) : x의 주소에 있는 값.

동적할당 사용법

동적할당 사용법

int Size, *arr;
scanf (“%d”, &Size);

Size 개의 원소를 갖는 int형 배열을 만들고 싶다.

arr = (int*) malloc (sizeof(int)*Size);

Size개의 int가 들어갈 만큼의 메모리를 할당받는다.
그리고 그 시작 주소를 arr에 집어넣는다!

프로그램이 끝날 때,
free (arr);을 해주는걸 잊지 말자.

동적할당 사용법

int Size, **arr;
scanf ("%d", &Size);

Size 개의 원소를 갖는 int*형 배열을 만들고 싶다.

arr = (int**) malloc (sizeof(int*)*Size);

Size개의 int*가 들어갈 만큼의 메모리를 할당받는다.
그리고 그 시작 주소를 arr에 집어넣는다!

프로그램이 끝날 때,
free (arr);을 해주는걸 잊지 말자.

동적할당 사용법

뭐가 바뀐지 찾았나요?

못찾았으면 돌아가서 확인!

동적할당 사용법

int *arr -> int **arr
sizeof (int) -> sizeof (int*)
(int*) -> (int**)

*규칙성을 찾았다!

동적할당 사용법

```
int *arr       ->  int **arr
sizeof (int) ->  sizeof (int*)
(int*)          ->  (int**)
```

동적 할당을 하려면,
배열의 원소보다 *이 하나 더 붙은
포인터를 써야 한다!

동적할당 사용법

엉엉 근데 1차원밖에 못하잖아

이딴걸 어따써

동적할당 사용법

int* -> 1차원 배열
int** -> 2차원 배열
int*** -> 3차원 배열

…
되는데요?

동적할당 사용법

2차원 배열은 뭘까?

[1차원 배열]의 배열!
2d_arr[100][10] 이 있다고 하면,
2d_arr[0] , 2d_arr[1], … , 2d_arr[99]
는 모두 10개의 원소를 갖는
1차원 배열이다!

동적할당 사용법

1차원 배열은 뭐다? int* !
int* 의 배열은 뭐였지?
int**로 했었지!

동적할당 사용법

```
int **arr, Size;
int i;
scanf (“%d”, &Size);
arr = (int**) malloc (sizeof (int*)*Size);

for (i=0; i<Size; i++)
        arr[i] = (int*) malloc (sizeof (int)*Size);
```

Size*Size 크기의 2차원 배열이 만들어졌다!

동적할당 사용법

3, 4, … , n 차원도 이런 식으로 할 수 있다!
이해 되지?
3차원은 2차원 배열의 배열, … 이런 느낌!

동적할당 사용법

주의! 감점 포인트>>>>>>

free()는 malloc()과 대칭으로 똑같이!
어떤 말이냐면..
arr = (int*) malloc (sizeof (int)*Size);
이랬으면
free (arr)만 하면 되지만..

동적할당 사용법

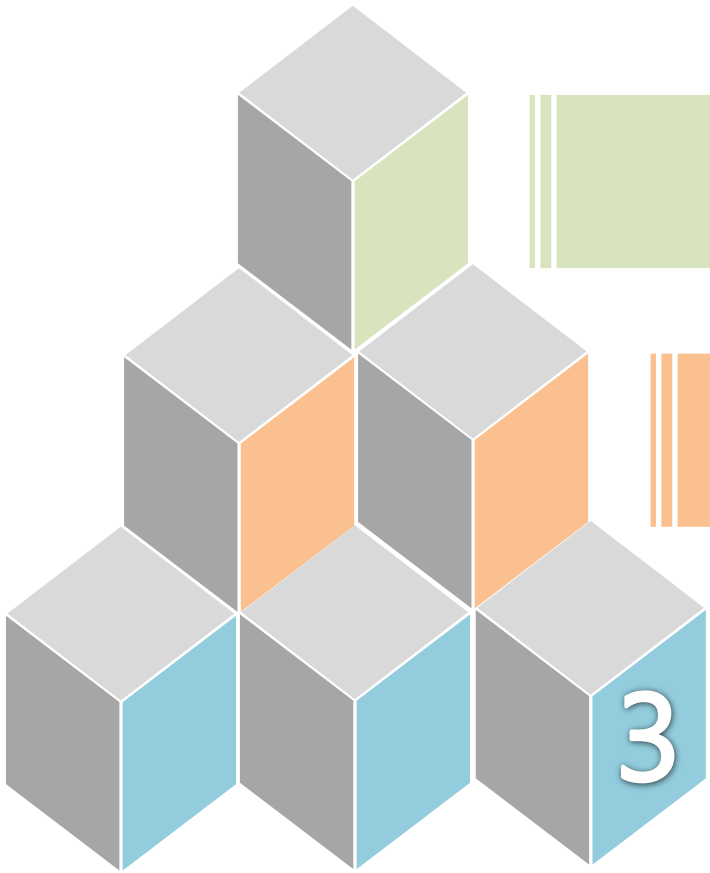
free()는 malloc()과 대칭으로 똑같이!

```
arr = (int**) malloc (sizeof (int*)*Size);
for (i=0; i<Size; i++)
        arr[i] = (int*) malloc (sizeof (int)*Size);
```

했으면..!

```
for (i=0; i<Size; i++)
        free (arr[i]);
free (arr);
```

이래야 한다!!!



포인터와 동적할당

포인터와 동적할당

다시 포인터로 돌아가보자!

```
void swap (int *x, int *y) {
        int tmp = *y;
        *y = *x;
        *x = tmp;
}
```

잘 된다.

포인터와 동적할당

다시 포인터로 돌아가보자!

```
void swap_pointer (int *x, int *y) {
        int *tmp = y;
        y = x;
        x = tmp;
}
```

잘 되냐?

포인터와 동적할당

다시 포인터로 돌아가보자!

```
void swap_pointer (int *x, int *y) {
        int *tmp = y;
        y = x;
        x = tmp;
}
```

안 된다!!

포인터와 동적할당

다시 포인터로 돌아가보자!

```
int* make_array (int size) {
        int* arr = (int*) malloc (sizeof (int)*size);
        return arr;
}
```

잘 되냐?

포인터와 동적할당

다시 포인터로 돌아가보자!

```
int* make_array (int size) {
        int* arr = (int*) malloc (sizeof (int)*size);
        return arr;
}
```

잘 된다.

포인터와 동적할당

다시 포인터로 돌아가보자!

```
void make_array (int size, int *arr) {
        arr = (int*) malloc (sizeof (int)*size);
}
```

잘 되냐?

포인터와 동적할당

다시 포인터로 돌아가보자!

```
void make_array (int size, int *arr) {
        arr = (int*) malloc (sizeof (int)*size);
}
```

잘 되냐?

안 된다!!

포인터와 동적할당

안되는 두 경우의 **공통점**이 뭐였을까?

포인터와 동적할당

안되는 두 경우의 **공통점**이 뭐였을까?

포인터를 바꾸려고 했다!

포인터와 동적할당

헉.. 포인터는 바꿀 수 없나요?

포인터와 동적할당

헉.. 포인터는 바꿀 수 없나요?
<=> 헉.. 포인터의 포인터는 없나요?

포인터와 동적할당

헉.. 포인터는 바꿀 수 없나요?
<=> 헉.. 포인터의 포인터는 없나요?

당연히 있습니다.
int* 의 포인터는 int**이고,
int**의 포인터는 int***이고, ...

포인터와 동적할당

포인터는 자신에게 *을
하나 붙인 값을 바꿀 수 있다.

int의 값을 바꾸려면 int*을 써야 하고,
int*의 값을 바꾸려면 int**을 써야 하고, ...

이런 식이다.

포인터와 동적할당

다시 아까의 예로!

```
void make_array (int size, int** arr) {
        *arr = (int*) malloc (sizeof (int)*size);
}
```

되냐?

포인터와 동적할당

다시 아까의 예로!

```
void make_array (int size, int** arr) {
        *arr = (int*) malloc (sizeof (int)*size);
}
```

된다!!!!

포인터와 동적할당

다시 아까의 예로!

```
void swap_pointer (int **x, int** y) {
        int *tmp = *y;
        *y = *x;
        *x = tmp;
}
```

되냐?

포인터와 동적할당

다시 아까의 예로!

```
void swap_pointer (int **x, int** y) {
        int *tmp = *y;
        *y = *x;
        *x = tmp;
}
```

된다!!!!

끝이야

복습해라
계속해라

끝이야

감사