



C/C++

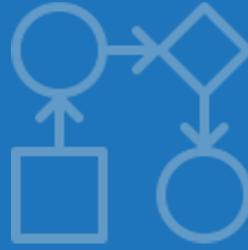
structure



pointer



function



switch/case



array[]



for, while



malloc/free

프로그래밍 기초

14장. 함수와 포인터

Lab. 이미지 처리 프로그램: 구조체 선언

■ 이미지 처리 함수 구현

- 흑백 영상의 픽셀값은 0~255 사이의 값을 가지며, 0은 검은색, 255는 흰색을 나타냄

■ main()에서 함수 호출 과정

- `main()` 함수
 - 2개의 구조체 IMAGE 변수를 선언 및 구조체 변수 초기화
 - `IMAGE image, backup;`
- `void fill_image(&image)` 함수 호출
 - 배열의 내용 출력
 - 구조체 변수 `image`의 값을 `backup` 구조체 변수에 복사함
 - `memcpy()` 사용
- `void convert_image(&image)` 함수 호출
 - 변경된 `buffer`의 내용 화면 출력
- `void reverse_image(&backup)` 함수 호출
 - 변경된 `buffer`의 내용 화면 출력

```
#define WIDTH 10
#define HEIGHT 10

typedef struct
{
    int width;
    int height;
    unsigned char buffer[WIDTH][HEIGHT];
} IMAGE;

int main(void)
{
    IMAGE image, backup;
    . . .
}
```

Lab. 이미지 처리 프로그램: 함수 구현

■ 함수 구현 내용

- `void fill_image(IMAGE *p)` 함수 구현
 - 파라미터로 전달된 image의 buffer[][]에 0~255 사이의 값을 랜덤하게 저장
- `void convert_image(IMAGE *p)` 함수 구현
 - 각 픽셀의 값(image[i][j])이 128 미만이면 0으로 만들고, 128이상이면 255을 만드는 함수 구현
 - 원본 이미지(image[][] 배열)을 함수 파라미터로 전달
- `void reverse_image(IMAGE *p)` 함수 구현
 - 255에서 현재 이미지의 pixel 값을 뺀
 - backup_image[][] 배열을 함수 파라미터로 전달
- `void print_image(IMAGE *p)` 함수 구현
 - 구조체 IMAGE 내부의 buffer[][]의 값을 출력
 - 자리 수 맞춤

실행 결과

Original Image									
232	19	158	38	175	197	114	68	188	109
120	80	102	47	102	143	142	79	160	52
3	124	114	32	70	18	189	123	116	190
247	56	17	157	230	3	139	79	204	66
22	167	208	141	155	125	158	16	54	157
56	53	118	49	163	35	84	116	30	193
22	211	24	89	251	223	42	123	228	43
13	211	160	178	15	154	233	126	200	14
30	19	234	106	32	185	15	104	6	228
183	173	125	202	177	131	16	162	158	159

Backup Image									
232	19	158	38	175	197	114	68	188	109
120	80	102	47	102	143	142	79	160	52
3	124	114	32	70	18	189	123	116	190
247	56	17	157	230	3	139	79	204	66
22	167	208	141	155	125	158	16	54	157
56	53	118	49	163	35	84	116	30	193
22	211	24	89	251	223	42	123	228	43
13	211	160	178	15	154	233	126	200	14
30	19	234	106	32	185	15	104	6	228
183	173	125	202	177	131	16	162	158	159

Converted Image(from image)									
255	0	255	0	255	255	0	0	255	0
0	0	0	0	0	255	255	0	255	0
0	0	0	0	0	0	255	0	0	255
255	0	0	255	255	0	255	0	255	0
0	255	255	255	255	0	255	0	0	255
0	0	0	0	255	0	0	0	0	255
0	255	0	0	255	255	0	0	255	0
0	255	255	255	0	255	255	0	255	0
0	0	255	0	0	255	0	0	0	255
255	255	0	255	255	255	0	255	255	255

Reversed Image(from backup)									
23	236	97	217	80	58	141	187	67	146
135	175	153	208	153	112	113	176	95	203
252	131	141	223	185	237	66	132	139	65
8	199	238	98	25	252	116	176	51	189
233	88	47	114	100	130	97	239	201	98
199	202	137	206	92	220	171	139	225	62
233	44	231	166	4	32	213	132	27	212
242	44	95	77	240	101	22	129	55	241
225	236	21	149	223	70	240	151	249	27
72	82	130	53	78	124	239	93	97	96