

## PART 16-1. 동적 할당 함수

### ◆ 연습문제 ◆

1. 다음 용도에 맞게 저장 공간을 동적 할당하여 각 포인터에 연결하세요.

① 몸무게를 저장할 변수

double \*weight =

② 10과목의 점수를 저장할 배열

int \*scores =

③ 널문자를 포함하여 최대 80글자의 문자열을 저장할 char 배열

char \*string =

정답 : ① double \*weight = (double \*) malloc(sizeof(double));

② int \*scores = (int \*) malloc(10 \* sizeof(int));

③ char \*string = (char \*) malloc(80 \* sizeof(char));

해설 : 없음

2. 다음은 동적 할당이 제대로 되었는지 검사하고 사용이 끝난 동적 할당 영역을 반환하도록 빈 칸을 채워 보세요.

int \*max = (int \*) malloc(sizeof(int));

```
if(      ①      ) {  
    printf("메모리가 부족합니다.");  
    return 0;  
}
```

```
*max = 999;  
( ② )
```

정답 : ① max == 0 또는 max == NULL  
② free(max);

해설 : 없음

3. 하루 중 최고기온과 최저기온을 입력하여 일교차를 출력하세요. 단, 온도를 입력할 저장 공간은 동적으로 할당합니다.

실행 결과

최고기온과 최저기온을 입력하세요 : 30.8 21.6  
일교차는 9.2도입니다.

정답

```
#include <stdio.h>  
#include <stdlib.h>
```

```
int main(void) {  
    double *high, *low;  
  
    high = (double *) malloc(sizeof(double));  
    low = (double *) malloc(sizeof(double));
```

```
    if ((high == 0) || (low == 0)) {  
        printf("메모리가 부족합니다.\n");  
        exit(1);  
    }
```

```
    printf("최고기온과 최저기온을 입력하세요 : ");  
    scanf("%1f%1f", high, low);
```

```

printf("일교차는 %.1lf도 입니다.\n", *high - *low);

free(high);
free(low);

return 0;
}

```

해설 : 없음

4. 다음 코드에서 pa,pb,pc 중 동적 할당 영역의 크기가 가장 큰 것은?

```

int ary[5] = {1, 2, 3, 4, 5};
int *pa, *pb, *pc;
int i;

pa = (int *) malloc(sizeof(ary));

for (i = 0; i < 5; i++) {
    pa[i] = ary[i];
}

pb = (int *) calloc(pa[3], sizeof(int));
pc = (int *) realloc(NULL, pa[4]);

```

정답 : pa

해설 : 없음

5. 메모를 반복 입력하여 이어붙이는 프로그램을 작성하세요. 새로운 메모를 입력할 때마다 입력한 메모의 전체 길이에 맞게 동적 할당 영역을 조정하여 메모를 이어붙입니다. 한 번에 입력할 수 있는 메모의 길이는 30자로 제한하며 'end'를 입력하거나 사용 가능한 동적 할당 메모리가 없으면 입력을 끝냅니다.

실행 결과

메모 입력 : The  
메모 입력 : most beautiful thing  
메모 입력 : in  
메모 입력 : the world is...  
메모 입력 : end  
The most beautiful thing in the world is...

정답

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(void) {
    char temp[32];
    char *ps, *pt;
    int len = 1;

    ps = (char *) calloc(1, 1);

    while(1) {
        printf("메모 입력 : ");
        fgets(temp, sizeof(temp), stdin);
        temp([strlen(temp) - 1] = '\0');

        if (strcmp(temp, "end") == 0) {
            break;
        }
        len += strlen(temp) + 1;
        pt = (char*) realloc(ps, len);
```

```
    if (pt == NULL) {  
        printf("메모리 부족...\n");  
        break;  
    }  
    ps = pt;  
    strcat(ps, temp);  
    strcat(ps, "");  
}  
  
printf("%s\n", ps);  
free(ps);  
  
return 0;  
  
}
```

해설 : 없음