



실 습 문 제

제17장

1. [동적 메모리 할당]

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

int main( void )
{
    int *p;

    p = (int *)malloc( sizeof(int) );
    if( p == NULL )
    {
        printf("동적 메모리 할당 오류\n");
        return 1;
    }

    *p = 100;
    printf(" *p = %d\n", *p);
    free( p );

    return 0;
}
```

(a) 위의 프로그램을 컴파일하여 실행하고 그 결과를 기록하라.

실행결과

- (b) 10개의 정수를 저장할 수 있는 공간을 할당받아서 1부터 10까지를 저장하여 보라. 저장된 값을 다시 읽어서 평균값을 구해서 출력한다.
- (c) 10개의 실수를 저장할 수 있는 공간을 할당받아서 1.0부터 10.0까지를 저장하여 보라. 저장된 값을 다시 읽어서 평균값을 구해서 출력한다.
- (c) 50개의 문자를 저장할 수 있는 공간을 할당받아서 문자열 "Actions speak louder than words."를 저장하여 보라. 저장된 문자열을 다시 화면에 출력하여 보라.
- (d) malloc() 대신에 calloc()을 사용하여 10개의 정수를 저장할 수 있는 공간을 할당받은 후에 초기값을 화면에 출력하여 보라.
- (e) realloc()을 이용하여 기존의 메모리 할당의 크기를 2배로 증가시켜 보라.

2. [연결 리스트] 문자열이 저장되어 있는 연결 리스트를 만들어 본다.

```

#include <stdio.h>
#include <malloc.h>

struct NODE {
    char data[10];
    struct NODE *link;
};

int main( void )
{
    struct NODE *first, *p;

    p = malloc(sizeof(struct NODE));
    if( p == NULL )
    {
        printf("동적 메모리 할당 오류\n");
        return 1;
    }
    strcpy(p->data, "cat");
    p->link = NULL;
    first = p;
    free(p);

    return 0;
}

```

(a) 위의 프로그램을 컴파일하여 실행하고 그 결과를 기록하라.
실행결과

- (b) 두 번째 노드를 생성하여 "bat"라는 문자열을 저장하고, 첫 번째 노드의 뒤에 연결하라. 두 번째 노드의 link 필드에는 NULL값을 저장하여야 한다.
- (c) 본문의 insert_node() 함수를 이용하여 위의 프로그램을 다시 작성하여 보라.
- (d) 특정한 데이터 값을 가지는 노드를 탐색하여 delete_node() 함수를 이용하여 삭제하여 보라.
- (e) 연결 리스트가 저장하고 있는 모든 데이터 값을 출력하여 보라.