

PART 13 ► 프로그래밍 실습

1. 도서 목록 관리 프로그램

도서 목록 관리 프로그램을 작성하려고 한다. 각 도서는 저자, 도서명, 출판연도 정보를 저장하고 있다. 여기에서 저자, 도서명은 각 도서마다 다르므로 동적 할당을 이용하여 메모리를 할당한다.

- (1) n 개의 도서 목록을 읽어 들여 이를 출력하는 프로그램을 작성하라. 하나의 도서 정보는 세 개의 행으로 구성되어 있다고 가정한다. 첫 번째 행에는 저자명이 오고, 두 번째 행에는 도서명이, 세 번째 행에는 출판연도가 나타난다. 도서 목록에서 각 도서 정보는 하나의 빈 행으로 구별한다. 예컨대 2개의 도서에 대한 도서 목록은 다음과 같다.

```
Polya
How to solve it
1957

Kernighan and Ritchie
Programming Language C
1988
```

각 도서 정보는 저자명, 도서명, 출판연도 순으로 출력한다. 한 행에 하나의 도서 정보가 나타나도록 출력하되 각 필드는 쉼표로 구별한다. 예컨대 위 도서 목록은 다음과 같이 출력되어야 한다.

```
Polya, How to solve it, 1957
Kernighan and Ritchie, Programming Language C, 1988
```

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

```
#define MAXSZ 256
```

```
typedef struct {
    char *author;
    char *title;
    int year;
```

```

} Book;

void read_record(Book *bookptr);
void print_records(Book book[], int n);

int main()
{
    int n;
    char buf[MAXSZ];

    printf("입력할 도서 개수를 입력해 주세요. ");
    scanf("%d", &n);
    fgets(buf, MAXSZ, stdin);    // 줄바꿈문자까지 읽어들이
    Book book[n];
    printf("%d개의 도서 정보를 입력하세요.\n");
    for (int i = 0; i < n; i++)
        read_record(&book[i]);
    printf("----- 입력한 도서 정보 -----");
    printf("-----\n");
    print_records(book, n);
    printf("-----");
    printf("-----\n");

    return 0;
}

char *read_line()
{
    char buf[MAXSZ];
    char *line;
    int len;

    fgets(buf, MAXSZ, stdin);
    len = strlen(buf);
    line = (char *) malloc(len);
    buf[len-1] = '\0';
    strcpy(line, buf);

    return line;
}

```

```

void read_record(Book *p)
{
    char *line, buf[MAXSZ];
    line = read_line();
    if (line[0] == '\\0') {
        puts(line);
        free(line);
        line = read_line();
    }
    p->author = line;
    p->title = read_line();
    scanf("%d", &p->year);
    fgets(buf, MAXSZ, stdin);    // 줄바꿈문자까지 읽어들이
}

void print_records(Book book[], int n)
{
    for (int i = 0; i < n; i++) {
        Book *bp = &book[i];
        printf("%s, %s, %d\\n", bp->author, bp->title, bp->year);
    }
}

```

- (2) 출력 형식을 HTML 형식으로 바꾸어 출력하여라. 제목은 “내가 좋아하는 책들”이라고 하고 각 도서 목록은 의 로 나타낸다. 단 도서명 부분은 으로 강조하여 나타낸다. 위에서 예시한 도서 목록을 HTML로 바꾸어 출력하면 다음과 같다.

```

<html>
  <head><title>내가 좋아하는 책들</title></head>
  <body>
    <ul>
      <li>Polya, <em>How to solve it</em>, 1957</li>
      <li>Kernighan and Ritchie, <em>Programming Language C</em>, 1988</li>
    </ul>
  </body>
</html>

```

```

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

```

```

#define MAXSZ 256

typedef struct {
    char *author;
    char *title;
    int year;
} Book;

void read_record(Book *bookptr);
void print_html(Book book[], int n);

int main()
{
    int n;
    char buf[MAXSZ];

    printf("입력할 도서 개수를 입력해 주세요. ");
    scanf("%d", &n);
    fgets(buf, MAXSZ, stdin); // 줄바꿈문자까지 읽어들이
    Book book[n];
    printf("%d개의 도서 정보를 입력하세요.\n");
    for (int i = 0; i < n; i++)
        read_record(&book[i]);
    print_html(book, n);

    return 0;
}

char *read_line()
{
    char buf[MAXSZ];
    char *line;
    int len;

    fgets(buf, MAXSZ, stdin);
    len = strlen(buf);
    line = (char *) malloc(len);
    buf[len-1] = '\0';
    strcpy(line, buf);

    return line;
}

```

```

    }

    void read_record(Book *p)
    {
        char *line, buf[MAXSZ];
        line = read_line();
        if (line[0] == '\\0') {
            puts(line);
            free(line);
            line = read_line();
        }
        p->author = line;
        p->title = read_line();
        scanf("%d", &p->year);
        fgets(buf, MAXSZ, stdin);    // 줄바꿈문자까지 읽어들이
    }

    void print_html(Book book[], int n)
    {
        printf("<html>\n");
        printf("  <head><title>내가 좋아하는 책들</title></head>\n");
        printf("  <body>\n");
        printf("    <ul>\n");
        for (int i = 0; i < n; i++) {
            printf("      <li> ");
            Book *bp = &book[i];
            printf("%s, <em>%s</em>, %d", bp->author, bp->title, bp->year);
            printf("</li>\n");
        }
        printf("    </ul>\n");
        printf("  </body>\n");
        printf("</html>\n");
    }
}

```

- (3) 읽어 들인 도서 목록을 출판연도 내림차순으로 출력하는 프로그램을 작성하라. 출력 형식은 역시 HTML로 한다.

```

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

```

```

#define MAXSZ 256

typedef struct {
    char *author;
    char *title;
    int year;
} Book;

void read_record(Book *bookptr);
void print_html(Book book[], int n);
void sort_records(Book book[], int n);

int main()
{
    int n;
    char buf[MAXSZ];

    printf("입력할 도서 개수를 입력해 주세요. ");
    scanf("%d", &n);
    fgets(buf, MAXSZ, stdin);    // 줄바꿈문자까지 읽어들이
    Book book[n];
    printf("%d개의 도서 정보를 입력하세요.\n");
    for (int i = 0; i < n; i++)
        read_record(&book[i]);
    sort_records(book, n);
    print_html(book, n);

    return 0;
}

char *read_line()
{
    char buf[MAXSZ];
    char *line;
    int len;

    fgets(buf, MAXSZ, stdin);
    len = strlen(buf);
    line = (char *) malloc(len);
    buf[len-1] = '\0';
    strcpy(line, buf);
}

```

```

        return line;
    }

void read_record(Book *p)
{
    char *line, buf[MAXSZ];
    line = read_line();
    if (line[0] == '\\0') {
        puts(line);
        free(line);
        line = read_line();
    }
    p->author = line;
    p->title = read_line();
    scanf("%d", &p->year);
    fgets(buf, MAXSZ, stdin);    // 줄바꿈문자까지 읽어들이
}

int cmp(const void *p, const void *q)
{
    Book *bk1 = (Book *)p, *bk2 = (Book *)q;
    return bk2->year - bk1->year;
}

void sort_records(Book *books, int n)
{
    qsort(books, n, sizeof(Book), cmp);
}

void print_html(Book book[], int n)
{
    printf("<html>\n");
    printf("  <head><title>내가 좋아하는 책들</title></head>\n");
    printf("  <body>\n");
    printf("    <ul>\n");
    for (int i = 0; i < n; i++) {
        printf("      <li> ");
        Book *bp = &book[i];
        printf("%s, <em>%s</em>, %d", bp->author, bp->title, bp->year);
        printf("</li>\n");
    }
}

```

```

    }
    printf("    </ul>\n");
    printf(" </body>\n");
    printf("</html>\n");
}

```

2. 삽입정렬 프로그램

연결 리스트를 이용하여 삽입정렬(insertion sort)을 수행하는 프로그램을 작성하라. 삽입정렬이란 데이터를 하나씩 읽을 때마다 제 위치를 찾아서 넣어 주는 정렬 방식이다. 예를 들어, 다음과 같이 다섯 개의 숫자가 있다고 할 때 이들을 오름차순으로 정렬한다고 생각해 보자.

3 5 2 4 7

제일 처음 3을 읽은 후에는 3 하나를 연결 리스트에 넣는다. 3 하나가 들어가 있는 상태는 그 자체로 정렬된 것이다. 그 다음에 5를 읽은 다음에는 3 다음 위치에 넣어야 한다. 그러면 연결 리스트에는 3, 5가 차례로 들어간 상태가 될 것이다. 이런 방식으로 연결 리스트에 삽입하였을 때, 그 과정을 순서대로 나타내면 다음과 같다.

3
 3 5
2 3 5
 2 3 4 5
 2 3 4 5 7

이런 방식으로 임의의 숫자 n개를 입력받은 후에 이들을 오름차순으로 정렬하는 프로그램을 작성하라. n개의 숫자가 모두 입력되면 EOF가 나타난다고 가정하자.

```

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

typedef struct _node {
    int data;
    struct _node *next;
} Node;

void insert(Node **list, Node *n);

```



```

void print(Node *list);

int main()
{
    Node *list = NULL;
    int n;

    while (scanf("%d", &n) == 1) {
        Node *p = (Node *)malloc(sizeof *p);
        p->data = n;
        p->next = NULL;
        insert(&list, p);
        print(list);
    }

    return 0;
}

void insert(Node **list, Node *n)
{
    if (*list == NULL)
        *list = n;
    else if ((*list)->data > n->data) {
        n->next = *list;
        *list = n;
    }
    else {
        insert(&((*list)->next), n);
    }
}

void print(Node *list)
{
    if (list == NULL)
        printf("\n");
    else {
        printf("%d ", list->data);
        print(list->next);
    }
}

```