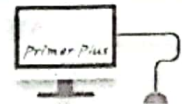




프로그래밍 연습



01 리스트 10.7의 강우량 프로그램을 수정하라. 계산하는 데 인덱스 대신 포인터를 사용하라. (여전히 배열을 선언하고 초기화시켜야 한다.)

02 double형의 배열을 초기화하고 그 배열의 내용을 다른 세 배열에 복사하는 프로그램을 작성하라. (네 개의 배열을 모두 메인 프로그램에서 선언해야 한다.) 첫 번째 복사를 위해서는 배열 표기를 가지는 함수를 사용하라. 두 번째 복사를 위해서는 포인터 표기와 포인터 증가를 가지는 함수를 사용하라. 처음 두 함수로 타겟 배열의 이름, 소스 배열의 이름, 복사할 원소의 개수를 복사되도록 하라. 세 번째 함수는 타겟의 이름, 소스의 이름, 소스의 마지막 원소 뒤에 오는 원소에 대한 포인터를 전달인자로 삼게 하라. 즉, 선언이 다음과 같이 주어졌을 때 각각의 함수 호출은 다음과 같아야 한다.

```
double source[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
double target1[5];
double target2[5];
double target3[5];
copy_arr(target1, source, 5);
copy_ptr(target2, source, 5);

copy_ptr(target3, source, source + 5);
```

03 int형의 배열에 저장된 값 중에서 가장 큰 값을 리턴하는 함수를 작성하고, 간단한 프로그램으로 그 함수를 테스트하라.

04 double형의 배열에 저장된 값 중 가장 큰 값의 인덱스를 리턴하는 함수를 작성하고, 간단한 프로그램으로 그 함수를 테스트하라.

05 double형의 배열에 저장된 값 중에서 가장 큰 값과 가장 작은 값의 차를 리턴하는 함수를 작성하고, 간단한 프로그램으로 그 함수를 테스트하라.

- 06 double형의 배열의 내용들을 반전시키는 함수를 작성하고 간단한 프로그램으로 그 함수를 테스트하라.
- 07 double형의 2차원 배열을 초기화시키고, 문제 2에서 작성한 복사 함수 중 하나를 사용하여 또 다른 2차원 배열에 복사하는 프로그램을 작성하라. (2차원 배열은 배열의 배열이기 때문에, 각각의 하위 배열에 1차원 복사 함수를 사용할 수 있다.)
- 08 문제 02에서 작성한 복사 함수를 사용하여, 원소 7개짜리 배열의 세 번째 원소부터 다섯 번째 원소까지를 원소 3개짜리 배열에 복사하라. 함수 자체를 수정할 필요는 없다. 실전달인자만 제대로 선택하면 된다. (실전달인자가 꼭 배열 이름과 배열 크기일 필요는 없다. 실전달인자로 배열 원소의 주소와 처리될 원소의 개수를 사용해야 한다.)
- 09 double형의 2차원 3×5 배열을 초기화하고, VLA에 기반을 둔 함수를 사용하여 또 다른 2차원 배열에 복사하는 프로그램을 작성하라. 두 배열의 내용을 VLA에 기반을 둔 함수로 출력해야 한다. 일반적으로, 두 함수는 임의의 $N \times M$ 배열을 처리할 수 있어야 한다. (사용하는 컴파일러가 VLA 기능을 지원하지 않는다면, $N \times 5$ 배열을 처리할 수 있는 전통적인 방식의 함수를 작성하라.)
- 10 두 배열에서 서로 대응하는 원소들의 합을 또 다른 배열에 원소로 저장하는 함수를 작성하라. 예를 들어, 배열 1이 2, 4, 5, 8 값을 가지고, 배열 2가 1, 0, 4, 6을 가진다면, 그 함수는 배열 3에 3, 4, 9, 14를 저장해야 한다. 그 함수는 세 개의 배열 이름과 하나의 배열 크기를 전달인자로 전달받아야 한다. 간단한 프로그램으로 그 함수를 테스트하라.
- 11 3×5 배열을 선언하고 적당한 값으로 초기화시키는 프로그램을 작성하라. 프로그램은 배열의 값들을 출력하고, 2배로 곱한 다음, 새로운 값을 출력해야 한다. 배열의 내용을 표시하는 함수와, 배열을 2배로 곱하는 함수를 따로 작성해야 한다. 그 함수들은 배열 이름과 행의 개수를 전달인자로 사용해야 한다.



12 리스트 10.7의 `rain.c` 프로그램을 다시 작성하라. 이번에는 필요한 작업들을 `main()`이 아니라 함수들을 사용하여 처리하라.

13 5개의 `double`형 값을 세 번 입력하도록 요구하는 프로그램을 작성하라. 프로그램은 다음과 같은 사항들을 처리해야 한다.

- a. 정보를 3×5 배열에 저장하라.
- b. 5개의 값으로 이루어진 각 집합에 대해 평균값을 구하라.
- c. 전체 값들에 대해 평균값을 구하라.
- d. 15개의 값 중에서 최대값을 구하라.
- e. 그 결과를 출력하라.

각각의 주요 작업은, C의 전통적인 배열 처리 방식을 사용하는 개별적인 함수로 처리해야 한다. 작업 "b"는, 1차원 배열의 평균값을 계산하고 리턴하는 함수를 사용하여 처리하라. 이 함수를 세 번 호출하는 루프를 사용하라. 그 밖의 작업들은 전체 배열을 전달인자로 전달하라. 작업 "c"와 "d"를 수행하는 함수는 호출 프로그램에 결과를 리턴해야 한다.

14 가변 길이 배열을 함수 매개변수로 사용하여, 프로그래밍 연습 13번을 다시 하라.

05

1에서 10까지 범위의 난수를 100개 발생시켜 내림차순으로 정렬하여 출력하는 프로그램을 작성하라. (11장: 문자열과 문자열 함수에서 보았던 정렬 알고리즘을 int형에 적용할 수 있다. 이 경우에는 수들을 정렬하면 된다.)

06

1에서 10까지 범위의 난수를 1,000개 발생시키는 프로그램을 작성하라. 난수들을 저장하거나 출력하지 마라. 다만 각 난수가 몇 번씩 발생하는지 출력하라. 서로 다른 10개의 종자값으로 프로그램을 테스트하라. 각 난수들이 같은 횟수로 발생하는가? 이 장에서 배운 함수들을 사용하거나, 같은 형식을 따르는 ANSI C의 rand()와 srand() 함수를 사용할 수 있다. 이것은 난수 발생기의 무작위성을 테스트하는 한 가지 방법이다.

07

리스트 12.13의 출력을 보여 준 뒤에 논의한 내용처럼 행동하는 프로그램을 작성하라. 즉, 프로그램의 출력이 다음과 같아야 한다.

던질 횟수를 입력하십시오(끝내려면 q).

18

주사위 면 수와 던질 주사위 개수를 입력하십시오(끝내려면 0).

6 3

6면 주사위 3개를 18번 던져 얻은 결과는 다음과 같습니다.

12 10 6 9 8 14 8 15 9 14 12 17 11 7 10

13 8 14

던질 횟수를 입력하십시오(끝내려면 q).

q

행운을 빕니다!

08

다음은 어떤 프로그램의 일부이다.

```
// pe12-8.c
#include <stdio.h>
int * make_array(int elem, int val);
void show_array(const int ar[], int n);
int main(void)
{
    int * pa;
    int size;
    int value;
```



```

printf("원소의 개수를 입력하십시오: ");
scanf("%d", &size);
while (scanf("%d", &size)==1 && size > 0)
{
    printf("초기화 값을 입력하십시오: ");

    scanf("%d", &value);
    pa = make_array(size, value);
    if (pa)
    {
        show_array(pa, size);
        free(pa);
    }
    printf("원소의 개수를 입력하십시오(끝내려면 <1): ");
}
printf("종료.\n");
return 0;
}

```

make_array()와 show_array()의 함수 정의를 작성하여 이 프로그램을 완성하라. make_array() 함수는 두 개의 전달인자를 사용한다. 첫 번째 전달인자는 int형 배열의 원소 개수를 나타내고, 두 번째 전달인자는 각 원소에 대입할 하나의 값을 나타낸다. 이 함수는 malloc()을 사용하여 적절한 크기의 배열을 생성하고, 지정된 값으로 각 원소들을 설정하고, 그 배열을 가리키는 포인터를 리턴한다. show_array() 함수는 배열의 내용을 출력하는데, 수들을 한 라인에 8개씩 출력한다.

09

다음과 같이 행동하는 프로그램을 작성하라. 먼저, 입력하려는 단어가 몇 개인지 물어본다. 그리고 나서 사용자가 단어를 입력하게 하고, 그런 다음 단어를 화면상에 나타낸다. malloc()과 첫 번째 질문(단어 수)에 맞는 답을 사용해 상응하는 수의 char를 가리키는 포인터(pointer-to-char)의 동적 배열을 만든다. (배열의 각 원소가 char를 가리키는 포인터(pointer-to-char)이기 때문에, malloc()의 리턴 값을 저장하는데 사용된 포인터는 char를 가리키는 포인터(pointers-to-a-pointer-to-char)를 가리키는 포인터이어야 함에 유의하라) 문자열을 읽을 때, 프로그램은 단어에 char의 임시 배열의 의미를 부여하여야 하며, malloc()를 사용해 단어를 유지하는데 충분한 저장 공간을 할당하고, char 포인터의 배열에 주소를 저장해야 한다. 이렇게,

문자 포인터의 배열 만들기를 마치고 각 포인터들은 특정 단어를 지정하기에 딱 알맞은 크기에 객체를 가리킨다. 예제는 다음과 같이 보일 수 있도록 실행한다:

How many words do you wish to enter? 5

Enter 5 words now:

I enjoyed doing this exercise

Here are your words:

I

enjoyed

doing

this

exercise



프로그래밍 연습



- 01 리스트 13.1의 프로그램을 수정하라. 이번에는 명령행 전달인자를 사용하지 말고, 사용자에게 파일 이름을 사용자에게 요청하여 그 응답을 읽어라.
- 02 소스 파일과 타깃 파일의 이름을 명령행으로부터 제공받는 파일 복사 프로그램을 작성하라. 가능하면 표준 입출력과 바이너리 모드를 사용하라.
- 03 소스 파일과 타깃 파일의 이름을 사용자에게 입력하도록 안내하는 프로그램을 작성하라. 이 프로그램은 타깃 파일에 저장하는 모든 텍스트를 대문자로 변환하기 위해 `ctype.h` 계열의 `toupper()` 함수를 사용해야 한다. 표준 입출력과 텍스트 모드를 사용하라.
- 04 명령행에 제시하는 모든 파일들을 차례대로 화면에 표시하는 프로그램을 작성하라. `argc`를 사용하여 루프를 제어하라.
- 05 리스트 13.5의 프로그램을 수정하라. 이번에는 대화식 인터페이스 대신에 명령행 인터페이스를 사용하라.
- 06 명령행 전달인자를 사용하는 프로그램은 그것의 정확한 사용법을 사용자가 기억해야 한다. 리스트 13.2의 프로그램을 수정하라. 이번에는 명령행 전달인자를 사용하지 말고, 필요한 정보를 대화식으로 사용자에게 요청하라.
- 07 두 개의 파일을 여는 프로그램을 작성하라. 파일 이름들은 명령행 전달인자를 사용하여 얻거나, 사용자에게 입력하도록 요청할 수 있다.
 - a. 첫 번째 파일의 첫 번째 라인, 두 번째 파일의 첫 번째 라인, 첫 번째 파일의 두 번째 라인, 두 번째 파일의 두 번째 라인 이런 식으로 출력하되, 둘 중 더 긴 파일(라인 수를 더 많은 파일)의 마지막 라인까지 출력한다.

- b. 라인 번호가 같은 라인들을 동일한 라인에 출력하도록 프로그램을 수정하라.

- 08 하나의 문자와 파일 이름들을 명령행 전달인자로부터 전달받는 프로그램을 작성하라. 문자 뒤에 파일 이름을 전혀 전달하지 않는다면, 프로그램은 표준 입력으로부터 읽어야 한다. 그렇지 않은 경우에는, 각 파일들을 차례로 열고, 각 파일에서 그 문자가 몇 번 나타나는지 보고해야 한다. 파일 이름과 문자를, 그 문자가 나타난 횟수와 함께 출력해야 한다. 전달인자의 개수가 정확한지 그리고 파일들이 제대로 열렸는지 알려 주는 에러 체크 코드를 포함시켜라. 어떤 파일을 열 수 없다면, 프로그램은 그 사실을 출력하고, 그 다음 파일로 넘어가야 한다.
- 09 리스트 13.3의 프로그램을 수정하라. 이번에는 목록에 추가되는 순서대로 각 단어에 1부터 번호를 붙여라. 프로그램이 다시 실행될 때 새로운 단어 넘버링은 이전의 단어 넘버링에 이어져야 한다.
- 10 사용자가 대화식으로 파일 이름을 제공한 텍스트 파일을 여는 프로그램을 작성하라. 사용자에게 파일 내의 위치를 입력하도록 요청하는 루프를 설정하라. 그러면 프로그램은 그 위치로부터 시작하여 다음 개행 문자가 나타날 때까지의 파일 내용을 출력해야 한다. 수치가 아닌 값을 입력하면 사용자 입력 루프가 종료되어야 한다.
- 11 두 개의 명령행 전달인자를 사용하는 프로그램을 작성하라. 첫 번째 명령행 전달인자는 문자열이다. 두 번째 명령행 전달인자는 파일 이름이다. 프로그램은 그 파일에서 그 문자열이 있는 모든 라인들을 출력해야 한다. 이 작업은 문자 지향적이 아니라 라인 지향적이기 때문에, `getc()` 대신 `fgetc()`를 사용하라. 그 문자열이 들어 있는 라인들을 찾기 위해, (11장: 문자열과 문자열 함수의 프로그래밍 연습 07에서 간단하게 사용한 적이 있는) 표준 C 라이브러리 함수 `strstr()`을 사용하라. 어떤 행도 문자 255개 보다 길지 않다고 가정하라.

12

30개의 정수로 이루어진 20개의 행을 가지고 있는 텍스트 파일을 작성하라. 그 정수들은 0부터 9까지의 범위이며, 스페이스로 분리되어 있다. 이 파일은 어떤 그림의 디지털 표현이다. 0부터 9까지의 값들은 명암을 나타내는데, 값이 클수록 어둡다. 파일의 내용을 int형의 20×30 배열 안으로 읽는 프로그램을 작성하라.

이 프로그램은 이 디지털 표현을 대강의 그림으로 변환한다. 프로그램은 배열에 있는 값들을 사용하여, 값 0은 스페이스 문자로, 값 1은 마침표 문자로 이런 식으로 값이 클수록 검은 공간을 더 많이 차지하는 문자로 각각 대체하여 char형의 20×31 배열을 초기화한다. 예를 들어, 값 9는 # 문자로 나타낼 수 있다. 각 행의 마지막 문자(31번째)는 20개의 행이 각각 문자열이 될 수 있도록 널 문자가 되어야 한다. 프로그램은 결과로 얻어지는 그림(즉, 문자열들)을 출력해야 한다. 또한 그 결과를 텍스트 파일에 저장해야 한다. 예를 들어, 다음과 같은 데이터를 가지고 시작한다고 가정하자.

```
0090000000000589985200000000000
0000900000000589985520000000000
0000000000000581985452000000000
0000900000000589985045200000000
0090000000000589985004520000000
0000000000000589185000452000000
0000000000000589985000045200000
55555555555558998555555555555
8888888888888589985888888888888
99990999999999999999999999999
8888888888888589985888888888888
55555555555558998555555555555
0000000000000589985000000000000
0000000000000589985000066000000
0000220000000589985005600650000
0000330000000589985056111165000
0000440000000589985005600650000
0000550000000589985000066000000
0000000000000589985000000000000
0000000000000589985000000000000
```

출력 문자들을 적절히 선택했을 때, 결과는 다음과 같을 것이다.

```
#          *%##%*1
#          *%##%*1
          *%.#%*~*1
#          *%##%* ~*1
#          *%##%* ~*1
          *%#.#%* ~*1
          *%##%* ~*1

*****%##%*****
%%%%%%%%%%%%%*%##%*%%%%%%%%%%%%%
#### #####:#####
%%%%%%%%%%%%%*%##%*%%%%%%%%%%%%%
*****%##%*****

          *%##%*
          *%##%* ==
' '      *%##%* *= *=
: :      *%##%* *= . . . =*
~ ~      *%##%* *= *=
**       *%##%* ==
          *%##%*
          *%##%*
```

- 13 위 프로그래밍 연습문제 12번을 표준 배열 대신 가변 길이 배열(VLAs)을 사용하여 하라.
- 14 디지털 이미지들은(특히 우주선으로부터 전송받는 이미지들은) 왜곡이 포함되어 있을 수 있다. 프로그래밍 연습 12에 왜곡을 보정하는 함수를 추가 하라. 이 함수는 각 값을 인접한 상하좌우의 값들과 비교해야 한다. 그 값이 상하좌우의 어느 값과 1 이상 차이가 나면, 상하좌우 값들의 평균값으로 그 값을 대체한다. 평균값은 가장 가까운 정수값으로 반올림해야 한다. 경계에 놓여 있는 값들은 이웃이 4개보다 적으므로, 특별한 조작이 필요하다.