

## 프로그래밍 연습문제 01

1. 입력으로 하나의 양의 정수를 받은 후 0이 될 때까지 연속적으로 2로 나눈 몫을 출력하는 프로그램을 작성하라. 예를 들어 입력된 정수가 51이면 25, 12, 6, 3, 1, 0을 순서대로 출력하면 된다. 입력된 정수가 127이면 63, 31, 15, 7, 3, 1, 0을 출력한다.

입력 예	출력
51	25 12 6 3 1 0
127	63 31 15 7 3 1 0
26	13 6 3 1 0
1	0

2. 먼저 입력될 정수의 개수  $2 \leq n \leq 100$ 을 입력받고, 이어서  $n$ 개의 정수를 입력받는다. 입력된 정수들 중에서 최소값과 두번째로 작은 값을 찾아 출력하는 프로그램을 작성하라. 만약 최소값이 2개 이상 중복되어 존재하면 그 중 하나를 최소값으로, 다른 하나를 두 번째로 작은 값으로 간주한다.

입력 예	출력
8 -7 -6 -12 -3 -7 -12 -1 -12	-12 -12
10 -5 8 0 -9 1 10 7 -7 0 9	-9 -7
8 7 2 1 0 8 9 9 3	0 1
2 7 7	7 7

3. 키보드로 부터 연속해서 음이 아닌 정수들을 입력받는다. 정수가 하나 씩 입력될 때 마다 현재까지 입력된 정수들을 오름차순으로 정렬하여 화면에 출력한다. 단, 새로 입력된 정수가 이미 이전에 입력된 정수라면 “duplicate”라고 출력하고 저장하지 않고 버린다. 사용자가 -1을 입력하면 프로그램을 종료한다. 입력되는 정수의 개수는 100개를 넘지 않는다.

입력 예	출력
5	5
2	2 5
5	duplicate
1	1 2 5
3	1 2 3 5
-1	
4	4
3	3 4
4	duplicate
45	3 4 45
12	3 4 12 45
5	3 4 5 12 45
4	duplicate
45	duplicate
2	2 3 4 5 12 45
1	1 2 3 4 5 12 45
7	1 2 3 4 5 7 12 45
-1	

4. 먼저 입력될 정수의 개수  $n \leq 100$ 을 입력받고, 이어서  $n$ 개의 정수를 받아 순서대로 배열에 저장한다. 그런 다음 키보드로 부터 다시 하나의 정수  $k$ 를 입력받은 후  $n$ 개의 정수들 중에서  $k$ 에 가장 가까운, 즉  $k$ 와의 차이의 절대값이 가장 작은 정수를 찾아 출력하는 프로그램을 작성하라.

입력 예	출력
6 1 9 8 2 5 12 7	8
6 -5 -1 -4 -1 -12 -9 20	-1
8 1 2 3 4 5 6 7 8 5	5
10 1 4 0 4 4 -11 4 1 1 6 -8	-11
12 73 28 1 9 37 46 -92 -8 37 0 0 12 24	28

5. 사용자로부터  $n < 100$ 개의 정수를 입력받아 크기순으로 정렬한 후 중복된 수를 제거하는 프로그램을 작성하라. 입력 형식은 먼저  $n$ 의 값이 주어지고 이어서  $n$ 개의 정수들이 주어진다. 예를 들어  $n = 8$ 이고 입력된 정수들이 4, 7, 4, 12, 4, 10, 9, 7 이라면 중복을 제거하고 남은 정수들은 4, 7, 9, 10, 12 이다. 그러면 먼저 남은 정수의 개수 5를 출력하고 콜론(:)을 출력한 후 남은 정수들을 오름차순으로 출력한다.

입력 예	출력
8 4 7 4 12 4 10 9 7	5: 4 7 9 10 12
5 1 1 1 1 1	1: 1

6. 주사위를  $N$ 번 던져서 1이 적어도  $k$ 번 나오는 확률을 실험적으로 계산하는 함수

```
double prob(int N, int k)
```

를 작성하라. 이 함수에서는 주사위를  $N$ 번 던져서 그 중 1이  $k$ 번 이상 나오는지는 검사하는 실험을 1,000,000번 반복한다. 이 1,000,000번의 실험 중에서 1이  $k$ 번 이상 나온 횟수를 카운트하여 확률을 실험적으로 추정한다. 이 함수를 이용하여 “ $k = 1, 2, \dots, 6$ 에 대해서 주사위를 6 $k$ 번 던져서 1이  $k$ 번 이상 나올 확률”이 각각 어떻게 되는지 계산하여 출력하는 프로그램을 작성하라. 6번 던져서 1이 1번 이상 나오는 사건의 확률은  $1 - (5/6)^6 \approx 0.6651$ 이고, 12번 중 적어도 2번 1이 나오는 사건의 확률은  $1 - (5/6)^{12} - 2(5/6)^{11} \approx 0.618667$ 이다. 이 두 경우에 대해서 대략 이 확률에 근접하는 결과가 나오면 정답이다. 참고로 C언어에서 랜덤 정수는 다음의 예에서와 같이 생성한다.

```
#include <stdio.h>
#include <stdlib.h>          // stdlib.h를 include한다.
#include <time.h>            // time.h를 include한다.

int main(void) {
    ...
    srand((unsigned int) time(NULL)); // main함수에서 이렇게 srand()를 실행한다.
    // srand는 프로그램 전체에서 한 번만 실행한다.
    ...

    int r = rand() % k;        // rand() 함수의 실행 결과를 양의 정수 k로 나누어 나머지
    // 를 구하면 0에서 k-1사이의 정수가 생성된다.
    ...
}
```

7. 입력으로  $n < 100$ 개의 영문 문자열을 받는다. 각 문자열의 길이는 20이하이다. 이 문자열들을 문자열의 길이가 짧은 것에서부터 긴 것 순서로 정렬하여 출력하라. 단, 길이가 동일한 문자열들은 그들 끼리 사전식 순서로 정렬되어야 한다. 입력 형식은 먼저 문자열의 개수  $n$ 이 주어지고, 이어서  $n$ 개의 문자열이 한 줄에 하나씩 주어진다. 문자열을 입력받아 배열에 저장하는 일은 main 함수에서 하고, 정렬하는 일은 별도의 함수를 만들어서 호출하라. 그런 다음 출력하는 일은 다시 main함수로 돌아와서 한다.

입력 예	출력
6 student university school department teacher college	school college student teacher department university
12 ca ab ba cdba abcdef dcb abc c a b bcda abcd	a b c ab ba ca abc dcb abcd bcda cdba abcdef

### 프로그래밍 연습문제 답안 제출방법

1. 문제 당 하나의 소스코드 파일을 제출한다. 소스파일의 이름은 prob\*.c 혹은 prob\*.cpp이다. 여기서 \*은 문제번호이다. 예를 들어 prob1.c, prob2.cpp 등이다. 이 파일들을 제외한 어떤 파일도 제출해서는 안된다.
2. 각 문제에 제시된 테스트 데이터를 모두 통과했는지 여부를 코드의 첫 줄에 코멘트의 형태로 기술한다. 이를 위반할 경우 미제출로 간주한다.
3. 각 문제에 대해서 입출력 요구사항을 정확하게 준수한다. 문제에서 요구하지 않은 불필요한 내용을 출력하거나 입력의 순서를 변경해서는 안된다.

## 프로그래밍 연습문제 02

- 영문 소문자로 구성된 2개의 단어를 입력받은 후 두 단어가 동일한 문자집합으로 구성되었는지를 검사하여 **yes** 혹은 **no**를 출력하는 프로그램을 작성하라. 예를 들어 **ababc**와 **cba**는 문자집합 {a, b, c}로 구성되었으므로 **yes**이다. 입력 단어의 길이는 20이하이다.

입력 예	출력
abababcccdabc cbaabba	no
there three	yes
strange stronger	no

- 입력으로 텍스트 파일 **harry.txt**를 읽어서 이 파일에 등장하는 모든 단어의 목록을 중복된 단어없이 사전식 순서로 정렬하여 강의 슬라이드 16페이지와 같은 자료구조로 저장한다. 이제 새로운 단어 하나를 키보드로부터 입력받는다. 저장된 단어들 중에 이 새로운 단어를 접두어(prefix)로 하는 모든 단어를 찾아서 한 줄에 하나씩 출력하는 프로그램을 작성하라. 마지막으로 찾아진 단어의 개수를 출력하라. 어떤 단어가 다른 단어의 접두어인지는 표준 라이브러리가 제공하는 **strncmp** 함수를 이용하여 검사할 수 있다. 이 함수의 사용 방법은 검색하여 알아보라.

접두어의 예	출력
str	straight straightening strain strange strangely stranger strawberry stray street stretch stretched strictly striding striking striped stripes strode stroke strolled strolling strong strongest struggled 23
help	help helped helping helplessly 4

접두어의 예	출력
there	there therefore 2

3. 입력으로 텍스트 파일 harry.txt를 읽는다. 이 텍스트 파일은 오직 영문 소문자만으로 구성되어 있다. 이 파일에 등장하는 길이가 6이상인 모든 단어의 목록과 각 단어의 등장 빈도를 구하여 화면으로 출력하는 프로그램을 작성하라. 단어들은 사전식 순서로 정렬되어 출력되어야 한다. 출력 파일의 각 줄에 하나의 단어와 그 단어의 등장 빈도를 콜론(:)으로 구분하여 출력하라. 동일한 단어가 중복해서 출력되어서는 안된다. 단어들은 강의 슬라이드 16페이지와 같은 자료구조로 저장되어야 한다. 출력의 마지막에는 전체 단어의 개수를 출력하라. 아래는 올바른 출력의 시작 부분과 끝 부분을 보여준다.

```

abandoning: 1
abroad: 1
absence: 1
accepting: 1
accidental: 2
accommodate: 1
accompanied: 1
according: 1
accusations: 1
achieved: 1
across: 14
...
yellow: 1
younger: 1
youthis: 1
zigzagged: 1
zoomed: 4
zooming: 2
1714

```

#### 프로그래밍 연습문제 답안 제출방법

1. 문제 당 하나의 소스코드 파일을 제출한다. 소스파일의 이름은 prob\*.c 혹은 prob\*.cpp이다. 여기서 \*은 문제번호이다. 예를 들어 prob1.c, prob2.cpp 등이다. 이 파일들을 제외한 어떤 파일도 제출해서는 안된다.
2. 각 문제에 제시된 테스트 데이터를 모두 통과했는지 여부를 코드의 첫 줄에 코멘트의 형태로 기술한다. 이를 위반할 경우 미제출로 간주한다.
3. 각 문제에 대해서 입출력 요구사항을 정확하게 준수한다. 문제에서 요구하지 않은 불필요한 내용을 출력하거나 입력의 순서를 변경해서는 안된다.

## 프로그래밍 연습문제 03

- 정렬(sort)을 하는 가장 간단한 방법 중의 하나는 다음과 같이 하는 것이다. 배열 `data`에 `data[0]`에서 `data[n-1]`까지  $n$ 개의 정수가 저장되어 있다. 먼저 `data[0] ~ data[n-1]` 사이의 정수들 중에서 가장 큰 정수를 찾는다. 그것을 `data[k]`라고 가정해보자. 그러면 `data[k]`와 `data[n-1]`을 교환(`swap`)한다. 이제 가장 큰 정수가 `data[n-1]`, 즉 맨 마지막 위치에 저장되었으므로 그 값에 대해서는 더 이상 생각할 필요가 없다. 이제 `data[0] ~ data[n-2]` 중에서 최대값을 찾는다. 그 값을 `data[p]`라고 하자. 그러면 다시 `data[p]`와 `data[n-2]`를 교환하고 `data[n-2]`에 대해서는 잊어버려도 된다. 이런 식으로 계속하면 마지막에는 `data[0]`와 `data[1]` 중에 최대값을 `data[1]`과 교환하면 전체의 정렬이 완료된다. 이 알고리즘을 구현하라. 입력은 먼저 정렬할 정수의 개수  $n \leq 100$ 이 주어지고 이어서  $n$ 개의 정수들이 주어진다.

입력 예	출력
8 4 7 4 12 4 10 9 7	4 4 4 7 7 9 10 12
5 1 1 1 2 2	1 1 1 2 2

- 입력으로  $n < 100$ 개의 구간(interval)이 주어진다. 각 구간은 구간의 시작점과 끝점으로 표현된다. 이 구간들을 시작점이 빠른 순서대로 정렬하여 출력하는 프로그램을 작성하라. 시작점이 같은 경우 끝점이 빠른 것을 먼저 출력한다. 입력 형식은 먼저  $n$ 의 값이 주어지고, 이어서 각 구간의 시작점과 끝점이 차례대로 주어진다. 각 구간의 시작점과 끝점은 정수이고, 끝점은 항상 시작점보다 크거나 같다.

입력 예	출력
8 // 구간의 개수	
-2 8 // 첫 번째 구간의 시작점과 끝점	-10 5
4 12 // 두 번째 구간의 시작점과 끝점	-2 8
4 7	4 7
4 10	4 10
-10 5	4 12
9 11	6 8
15 18	9 11
6 8 // 마지막 구간의 시작점과 끝점	15 18
3	-10000 -100
2000000 4000000	-10000 300000
-10000 300000	2000000 4000000
-10000 -100	

- 사전 파일 `shuffled_dict.txt`을 읽는다. 이 파일에는 각 줄마다 하나의 “단어”와 그 단어에 대한 “설명”이 저장되어 있다. “단어”와 그 단어에 대한 “설명”은 하나의 탭 문자(‘\t’)로 구분되어 있다. “단어”는 하나의 영문 소문자 문자열이며, “설명”은 여러 개의 단어로 구성된 문장이다. 이 사전 파일에서 단어들은 사전식 순서로 정렬되어 있지 않다. 이 파일을 읽은 후 단어들을 사전식 순서로 정렬하여 “`sorted_dict.txt`”라는 이름의 새로운 파일을 생성하여 저장하는 프로그램을 작성하라. 저장된 파일에서는 한 줄에 하나의 단어와 그 단어에 대한 설명을 탭 문자로 구분하여 저장해야 한다. 아래는 올바른 출력 파일의 시작과 끝 부분을 보여준다.

```
aam    a dutch and german measure of liquids varying in different cities
being at amsterdam about wine gallons at antwerp  at hamburg
abaca  the manilahemp plant musa textilis also its fiber see manila hemp
under manila
abacist      one who uses an abacus in casting accounts a calculator
...
```

zygosphenes a median process on the front part of the neural arch of the vertebrae of most snakes and some lizards which fits into a fossa called the zygantrum on the back part of the arch in front  
zymogene one of a physiological group of globular bacteria which produces fermentations of diverse nature distinguished from pathogene  
zymome a glutinous substance insoluble in alcohol resembling legumin now called vegetable fibrin vegetable albumin or gluten casein

### 프로그래밍 연습문제 답안 제출방법

1. 문제 당 하나의 소스코드 파일을 제출한다. 소스파일의 이름은 prob\*.c 혹은 prob\*.cpp이다. 여기서 \*은 문제번호이다. 예를 들어 prob1.c, prob2.cpp 등이다. 이 파일들을 제외한 어떤 파일도 제출해서는 안된다.
2. 각 문제에 제시된 테스트 데이터를 모두 통과했는지 여부를 코드의 첫 줄에 코멘트의 형태로 기술한다. 이를 위반할 경우 미제출로 간주한다.
3. 각 문제에 대해서 입출력 요구사항을 정확하게 준수한다. 문제에서 요구하지 않은 불필요한 내용을 출력하거나 입력의 순서를 변경해서는 안된다.

## 프로그래밍 연습문제 04

- 수열에서 큰 값이 작은 값보다 앞서 나오는 경우 두 값을 역전된(inverted) 쌍이라고 부른다. 예를 들어 수열 4, 2, 1, 1, 3에는 (4, 2), (4, 1), (4, 1), (4, 3), (2, 1), (2, 1)의 총 6개의 역전된 쌍이 있다. 수열을 입력으로 받아서 역전된 쌍의 개수를 카운트하여 출력하는 프로그램을 작성하라. 키보드로부터 먼저 정수의 개수  $N \leq 100$ 을 입력받고, 이어서  $N$ 개의 정수를 입력 받는다.

입력 예	출력
5 4 2 1 1 3	6
10 3 8 10 7 2 1 8 9 11 7	18
8 1 1 1 1 1 1 1 1	0
8 8 7 6 5 4 3 2 1	28

- 입력으로 하나의  $N \times N$  행렬이 저장된 파일 `input.txt`가 주어진다. 파일의 첫 줄에는 행렬의 크기  $N \leq 10$ 이 저장되어 있고, 이어진  $N$ 줄에는 각 줄마다  $N$ 개의 정수가 저장되어 있다. 이 파일을 읽어서 각 행의 평균과 표준편차, 각 열의 평균과 표준편차를 구해서 아래의 예와 같은 형식으로 화면으로 출력하는 프로그램을 작성하라. 평균과 표준편차는 소수점 이하 2자리까지 출력한다. 세부적인 출력 형식은 아래의 예를 따른다. 참고로 표준편차는 다음과 같이 정의된다. 루트(square root)를 계산하기 위해서 `math.h`를 `include`하고 `sqrt`함수를 사용하라.

$$SD = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

입력 예	출력
4	1 4 19 4 7.00 7.03
1 4 19 4	-5 0 8 3 1.50 4.71
-5 0 8 3	2 8 2 2 3.50 2.59
2 8 2 2	0 2 8 4 3.50 2.95
0 2 8 4	-0.50 3.50 9.25 3.25
	2.69 2.95 6.13 0.82
8	12 -9 13 7 22 91 0 -81 6.87 43.86
12 -9 13 7 22 91 0 -81	9 0 -10 8 14 9 -71 91 6.25 41.12
9 0 -10 8 14 9 -71 91	7 44 -18 0 1 12 31 61 17.25 24.38
7 44 -18 0 1 12 31 61	0 0 -56 9 12 8 21 22 2.00 23.23
0 0 -56 9 12 8 21 22	-9 8 12 42 81 16 9 0 19.87 26.90
-9 8 12 42 81 16 9 0	-12 6 8 7 12 21 23 7 9.00 10.04
-12 6 8 7 12 21 23 7	7 12 9 22 -7 -9 0 26 7.50 11.80
7 12 9 22 -7 -9 0 26	-18 17 0 11 2 8 11 16 5.87 10.62
-18 17 0 11 2 8 11 16	-0.50 9.75 -5.25 13.25 17.12 19.50 3.00 17.75
	10.42 14.95 21.78 12.28 25.58 28.21 29.79 46.78

- 다음과 같은 형식의 입력 파일 `table.txt`를 읽어서 아래의 예와 같이 `output.txt`파일에 테이블 형태로 출력하는 프로그램을 작성하라. 입력 파일의 첫 줄에는 테이블의 행의 개수  $m \leq 10$ 과 열의 개수  $n \leq 10$ 이 주어진다. 이어진  $m$ 줄에는 각 줄마다 테이블의 한 행에 들어갈  $n$ 개의 내용이 문자 &로 구분되어 주어진다. 테이블의 어떤 칸은 빈 칸일 수도 있다는 것을 유념하라. 빈 칸인 경우에도 하나 이상의 공백 문자로 표현되어 있다. 불필요한 공백 문자들을 제거하여 최대한 깔끔하게 출력되도록 하라.

### [입력 파일 `table.txt`의 예]

```
4 4                                     // 4행 4열의 테이블임을 의미한다.
Audio & www.ti.com/audio             & Automotive and Transportation & www.ti.com/automotive
```



Amplifiers & Communications and Telecom & [www.ti.com/communications](http://www.ti.com/communications)  
Data Converters & [dataconverter.ti.com](http://dataconverter.ti.com) & [www.ti.com/computers](http://www.ti.com/computers)  
& [www.dlp.com](http://www.dlp.com) & Consumer Electronics & [www.ti.com/consumer-apps](http://www.ti.com/consumer-apps)

#### [출력 파일 output.txt의 예]

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>	Automotive and Transportation	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Amplifiers		Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>		<a href="http://www.ti.com/computers">www.ti.com/computers</a>
	<a href="http://www.dlp.com">www.dlp.com</a>	Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>

#### 프로그래밍 연습문제 답안 제출방법

1. 문제 당 하나의 소스코드 파일을 제출한다. 소스파일의 이름은 prob\*.c 혹은 prob\*.cpp이다. 여기서 \*은 문제번호이다. 예를 들어 prob1.c, prob2.cpp 등이다. 이 파일들을 제외한 어떤 파일도 제출해서는 안된다.
2. 각 문제에 제시된 테스트 데이터를 모두 통과했는지 여부를 코드의 첫 줄에 코멘트의 형태로 기술한다. 이를 위반할 경우 미제출로 간주한다.
3. 각 문제에 대해서 입출력 요구사항을 정확하게 준수한다. 문제에서 요구하지 않은 불필요한 내용을 출력하거나 입력의 순서를 변경해서는 안된다.

## 프로그래밍 연습문제 05

- 하나의 영문 소문자로 된 문자열이 입력으로 주어진다. 이 문자열에서 자음이 가장 여러 번 연속해서 등장하는 부분을 찾아서 그 부분을 출력하는 프로그램을 작성하라. 예를 들어 문자열 “nietzsche”에서는 “tzsch”가 가장 긴 연속된 자음이다. 입력 문자열의 길이는 100이하이고, ‘a’, ‘e’, ‘i’, ‘o’, ‘u’를 제외한 모든 알파벳은 자음으로 간주한다.

입력 예	출력
nietzsche	tzsch
afgsyefklasjadasdgggdhsgdjasghdjhgaajgsd	sdgggdhsgdj
eetghaaa <b>bbstr</b> uuugdjsadgjaasdg	bbstr

- 프로그램을 시작하면 먼저 텍스트 파일 data.mat을 읽는다. 이 파일의 첫 줄에는 양의 정수  $N \leq 100$ 이 주어지고, 이어진  $N$ 줄에는 각 줄마다  $N$ 개의 정수가 주어진다. 즉, 하나의  $N \times N$  정수 행렬이 주어진다. 다음은 입력 파일 data.mat의 예이다:

```
8
12 -5  8 10 17  8 -6 21
-9  0  0 17  6 12 12  3
17 23  9  0  6  9  4  1
28 73  6 42  8 46  2  8
 3  6  4 19 -8  2 -7  3
-2  3 -7  8  4  6 28 -3
 7 46 82  6 43 28 -7  6
17  8  0 -9 12  7 19  9
```

그런 다음 화면에 프롬프트(\$)를 출력하고 다음의 예와 같이 사용자 명령을 처리하는 프로그램을 작성하라.

```
$ show    // 행렬을 출력한다.
12 -5  8 10 17  8 -6 21
-9  0  0 17  6 12 12  3
17 23  9  0  6  9  4  1
28 73  6 42  8 46  2  8
 3  6  4 19 -8  2 -7  3
-2  3 -7  8  4  6 28 -3
 7 46 82  6 43 28 -7  6
17  8  0 -9 12  7 19  9

$ colmax  // 각 열의 최대값을 출력한다.
28 73 82 42 43 46 28 21
$ colmin  // 각 열의 최소값을 출력한다.
-9 -5 -7 -9 -8 2 -7 -3
$ rowmax  // 각 행의 최대값을 출력한다.
21 17 23 73 19 28 82 19
$ rowmin  // 각 행의 최소값을 출력한다.
-6 -9 0 2 -8 -7 -7 -9
$ slice 1 2 2 3  // 아래에 이 명령의 의미를 설명한다.
```

```

0 12
6 46
-7 6
0 7
$ slice 0 1 1 2
-5 10 8 21
0 17 12 3
23 0 9 1
73 42 46 8
6 19 -2 3
3 8 6 -3
46 6 28 6
8 -9 7 9
$ exit // 프로그램을 종료한다.

```

명령 slice는 4개의 매개변수를 받는데 그것을 순서대로 x, p, y, q라고 하자. 그러면 행렬의 세로 축으로는 x번 행(행과 열의 번호는 0에서 시작한다), x+p번 행, x+2p번 행, x+3p번 행, ... 을 선택하고, 가로 축으로는 y번 열, y+q번 열, y+2q번 열, ... 을 선택하여 출력한다. 즉, 위에서 첫 번째 slice 명령의 예에서는 세로 축으로는 1번 행에서 시작하여 한 행씩 건너뛰는 것이고, 가로 축으로는 2번 열에서 시작하여 두 열씩 건너뛰는 것이다. 두 번째 slice 명령에서는 0번 행에서 시작하여 어떤 행도 건너뛰지 않았고, 1번 열에서 시작하여 한 열씩 건너뛰는 것이다.

- 입력 파일 board.txt에 오목판의 상태가 주어진다. 파일의 첫 줄에는 바둑판의 크기  $N \leq 19$ 가 주어지고, 이어진 N줄에는 각 줄마다 N개의 정수 0, 1, 혹은 2가 주어진다. 0은 빈자리를 표시하고, 1은 검은 돌, 2는 흰 돌을 표시한다. 주어진 상태가 검은 돌이 이긴 상태인지, 흰 돌이 이긴 상태인지, 혹은 아직 아무도 못 이긴 상태인지 검사하여 Black, White, 혹은 Not Finished라고 출력하는 프로그램을 작성하라. 둘 다 이긴 상태는 없다고 가정한다. 참고로 오목 게임은 내 돌이 수평, 수직, 혹은 대각선 방향으로 연속해서 5개가 놓이면 이기는 게임이다.

입력 예	출력
16 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 2 0 2 0 0 0 0 0 0 0 0 0 0 2 0 1 1 1 2 2 0 2 0 0 0 0 0 0 0 1 1 0 1 0 1 0 2 0 0 0 0 0 0 0 0 2 2 1 1 1 0 2 0 1 0 0 0 0 0 0 0 0 1 0 1 0 2 0 0 0 0 0 0 0 0 0 0 2 0 1 2 2 0 0 0 0 0 0 0 0 0 0 0 0 2 0 1 0	White

입력 예	출력
10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 2 0 1 0 0 1 0 0 0 1 2 2 2 2 1 0 0 0 1 0 2 1 1 1 2 0 0 0 0 2 1 0 1 2 1 0 1 0 0 0 2 2 1 1 0 1 0 0 0 0 0 2 2 0 1 0 1 0 0 1 2 2 2 2 1 2 0 2 0 0 0 0 0 1 0 0 0 0	Not Finished
15 0 0 0 0 2 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 2 0 1 0 1 0 2 0 0 0 0 0 1 0 2 0 1 0 1 2 1 0 2 0 0 0 0 0 2 0 1 2 1 1 1 2 2 0 2 0 0 0 2 0 2 0 1 0 1 0 2 0 1 0 0 0 1 0 0 0 2 0 1 2 1 0 2 0 0 0 0 0 0 0 0 0 2 0 2 1 2 0 0 0 0 0 0 0 0 0 0 0 1 0 2 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0	Black

4. sample.html 파일을 읽어서 파일에 등장하는 모든 HTML 태그(tag)들을 제거한 후 sample.txt라는 이름의 파일로 저장하는 프로그램을 작성하라. HTML 태그란 ‘<’와 ‘>’ 문자로 둘러싸진 부분을 의미한다. 단, HTML 파일에서 태그를 제거하는 것 말고 다른 부분은 그대로 유지되어야 한다. 즉, 출력 파일 sample.txt는 correct\_answer.txt 파일과 완벽하게 동일해야 한다.

#### 프로그래밍 연습문제 답안 제출방법

- 문제 당 하나의 소스코드 파일을 제출한다. 소스파일의 이름은 prob\*.c 혹은 prob\*.cpp이다. 여기서 \*은 문제번호이다. 예를 들어 prob1.c, prob2.cpp 등이다. 이 파일들을 제외한 어떤 파일도 제출해서는 안된다.
- 각 문제에 제시된 테스트 데이터를 모두 통과했는지 여부를 코드의 첫 줄에 코멘트의 형태로 기술한다. 이를 위반할 경우 미제출로 간주한다.
- 각 문제에 대해서 입출력 요구사항을 정확하게 준수한다. 문제에서 요구하지 않은 불필요한 내용을 출력하거나 입력의 순서를 변경해서는 안된다.

## 프로그래밍 연습문제 06

입력으로 텍스트 파일 harry.txt를 읽는다. 이 텍스트 파일은 오직 영문 소문자만으로 구성되어 있다. 이 파일에서 다음과 같은 일을 순서대로 수행하는 프로그램을 작성하라(각각을 별개의 프로그램으로 만들지 말고 하나의 프로그램으로 구현하라.)

1. 입력 파일에 등장하는 모든 단어의 목록과 각 단어의 등장 빈도를 구하여 화면으로 출력하는 프로그램을 작성하라. 단어들은 사전식 순서로 정렬되어 출력되어야 한다. 출력 파일의 각 줄에 하나의 단어와 그 단어의 등장 빈도를 콜론(:)으로 구분하여 출력하라. 동일한 단어가 중복해서 출력되어서는 안된다. 출력의 마지막 에는 전체 단어의 개수를 출력하라. 아래는 올바른 출력의 시작 부분과 끝 부분을 보여준다. 단어들은 하나의 연결리스트로 저장되어야 한다. 주의: 파일로부터 단어들을 읽어서 먼저 배열에 모두 저장한 후 나중에 다시 연결리스트 리스트로 만드는 식으로 해서는 안된다. 단어들은 파일에서 읽는 즉시 연결리스트에 저장되어야 한다. 또한 중복 검사 없이 모든 단어를 연결리스트에 저장한 후 나중에 중복을 제거하는 식으로 해서는 안된다. 연결리스트에는 항상 단어들이 중복없이 저장되어 있도록 구현해야 한다.

```
a: 478
abandoning: 1
able: 6
ably: 1
about: 52
above: 2
abroad: 1
absence: 1
accepting: 1
accidental: 2
accommodate: 1
...
younger: 1
your: 31
yours: 2
youthis: 1
zards: 1
zigzagged: 1
zoomed: 4
zooming: 2
2763
```

2. 1번에서 만든 연결리스트에서 등장 빈도가 10 이하인 모든 단어들을 연결리스트로부터 제거한 후 남아 있는 단어들만 1번과 같은 형식으로 출력하라. 단어들은 사전식 순서로 정렬되어 있어야 하며, 마지막에 남아있는 단어들의 개수를 출력한다.

```
a: 478
about: 52
across: 14
all: 70
an: 45
and: 552
any: 20
anything: 16
are: 30
...
without: 11
wizard: 16
```

```
wizards: 15
world: 24
would: 42
you: 136
your: 31
211
```

3. 2번에서 만든 연결리스트에서 노드들을 단어들의 등장 빈도의 내림차순으로 정렬하라. 등장 빈도가 동일한 단어들은 사전식 순서로 정렬되어야 한다. 2번에서 만든 연결리스트에서 노드들을 하나씩 떼어서 새로운 정렬된 연결리스트에 삽입하는 방식으로 구현하라. 정렬된 연결리스트를 1번과 2번과 같은 형식으로 출력하라.

```
the: 992
and: 552
to: 519
a: 478
of: 429
he: 347
his: 294
was: 290
said: 268
had: 265
...
owl: 11
people: 11
school: 11
set: 11
three: 11
top: 11
while: 11
without: 11
211
```

### 프로그래밍 연습문제 답안 제출방법

1. 문제 당 하나의 소스코드 파일을 제출한다. 소스파일의 이름은 prob\*.c 혹은 prob\*.cpp이다. 여기서 \*은 문제번호이다. 예를 들어 prob1.c, prob2.cpp 등이다. 이 파일들을 제외한 어떤 파일도 제출해서는 안된다.
2. 각 문제에 제시된 테스트 데이터를 모두 통과했는지 여부를 코드의 첫 줄에 코멘트의 형태로 기술한다. 이를 위반할 경우 미제출로 간주한다.
3. 각 문제에 대해서 입출력 요구사항을 정확하게 준수한다. 문제에서 요구하지 않은 불필요한 내용을 출력하거나 입력의 순서를 변경해서는 안된다.