

[문제 1] 정수 N입력받고, N 줄의 공백을 포함하는 문자열을 입력받아, 가장 긴 문자열과 가장 짧은 문자열을 출력하고, 가장 긴 문자열의 가장 짧은 단어와 가장 짧은 문자열의 가장 긴 단어를 합성하여 하나의 단어로 만들어 출력하는 프로그램을 작성하시오. 단, 두 단어의 연결은 사전 순서상 빠른 단어 뒤에 순서가 늦은 단어를 연결한다.

- $N \leq 20$ 이고, 입력받은 문자열의 최대 길이는 100이고, 각 문자열에 포함된 단어의 최대 개수는 20이다.
- 첫 번째 단어의 위치는 0번부터 시작한다.
- 가장 긴 또는 가장 짧은 문자열이 두 개 이상인 경우, 먼저 입력된 문자열을 출력한다.

입력 예시 1

```
5      ↳ N은 줄 수
book desk pencil paper
orange apple banana lemon grape
red yellow green orange blue white black
candy cake icecream
july august september october november december
```

출력 예시 1

```
july august september october november december    ↳ 가장 긴 문자열
candy cake icecream                                  ↳ 가장 짧은 문자열
icecreamjuly
↳ 수정: 가장 긴 문자열에서 가장 짧은 단어와 가장 짧은 문자열에서 가장 긴 단어를 선택하여 사전 순서로 합성
```

[문제 2] 정수 N을 입력받고, 공백을 포함하지 않는 문자열 한 개를 입력받아 정적 배열 X에 저장한다. 다음, N 줄의 공백을 포함하는 문자열을 입력받아, 동적으로 저장 장소를 할당 받은 후 문자형 포인터 배열 Y에 연결한다. Y의 데이터 타입은 char **이며, 포인터 배열도 동적으로 할당을 받는다.

이 후, Y에 저장된 N 줄의 각 문자열에서 X에 저장된 문자열이 가장 많이 나타나는 문자열과 두 번째로 가장 많이 나타나는 문자열을 찾아 출력하는 프로그램을 작성 하시오. 단, 가장 많이 나타나는 문자열이 여러 개인 경우에는 가장 처음에 입력된 문자열을 출력한다.

- 입력되는 모든 문자열의 최대 길이는 100이며, 배열 X에 저장된 문자열의 길이는 Y에 저장된 문자열의 길이 보다 작거나 같다고 가정한다.
- 문자열 AAA에는 문자열 AA가 1개 포함되는 것으로 간주하며, 문자열 AAAA에는 문자열 AA가 2개 포함되는 것으로 간주한다. 즉, 이미 횟수 계산에 사용된 문자는 다음 계산에 포함하지 않는다.
- 배열 X에 저장된 문자열이 Y에 저장된 각 문자열에 한 번도 나타나지 않는 경우에는 NONE을 출력한다.

입력 예시 1

출력 예시 1

3 abc	
abcabc abcabc	↳ abc 4회
ABCabc ABCabc	↳ abc 2회
abc abc abcd	↳ abc 3회

abcabc abcabc	↳ 반복 횟수가 같은 문자열이 여러
	↳ 개 일 경우 먼저 입력된 문자열 출력
abc abc abcd	↳ 두 번째로 반복회수가 가장 많은

[문제 3] 다음의 작업을 하는 프로그램을 작성하시오.

(1) 공백을 포함하는 두 개의 문자열, 문자열 A와 문자열 B를 입력받고, 입력받은 문자열을 단어로 나누어 동적으로 할당받은 두 개의 2차원 배열(포인터 배열에 연결된 문자 배열)에 각각 저장한다. 단, 포인터 배열의 크기는 단어의 수와 같아야 하며, 포인터 배열에 연결되는 각 문자 배열은 단어의 길이+1의 크기여야 한다.

(2) 문자열 A에 포함된 두 개의 단어를 합성하여 만들어진 단어가 문자열 B에 포함되어있는 문자열 중 가장 긴 문자열을 출력한다. 합성단어의 길이가 동일한 경우 합성된 사전순서가 앞선 것을 출력한다.

- 문자열의 최대 크기는 100이다.
- 문자열의 공백과 공백 사이를 하나의 단어라고 본다. 문자열에 연속된 공백이 포함되지는 않는다. 문자열의 첫 번째 또는 마지막 문자가 공백인 경우는 없다.
- 문자열 A에 동일한 단어가 반복해서 나타나거나, 문자열 B에 동일한 단어가 반복해서 나타나는 경우는 없다.
- 문자열 A에 "or"와 "ange"의 단어가 있어 합성어 "orange"가 만들어지고,
- 문자열 A에 포함된 단어로 합성한 단어 중, 문자열 B에 포함되는 단어는 최소 1개 이상 있다.

입력 예시 1

red orange yellow green blue purple pink red redorange orange red yellow black bluegreen white purple
--

출력 예시 1

bluegreen ↳ bluegreen, orangered, redorange는 모두 동일한 길이, 이 중에서 사전순으로 가장 앞선 것은 bluegreen

[문제 4] 세종대 수시모집 전형은 2단계로 구성되어 있고, 이 중 2단계 전형에서는 1단계 점수 70%와 면접 점수 30%를 합산하여 **총점이 높은 학생 순으로 신입생을 선발한다**. 1단계 합격자는 N명이고, **최종 M%명의 신입생을 선발하며**, 각 지원자의 이름과 1단계 성적, 면접 점수가 아래 예시와 같이 주어질 때 **최종 합격자의 이름과 1단계 점수, 면접 점수, 총점을 출력하고, 최종 합격자의 수도 출력**하는 프로그램을 작성하시오. 동점자에 대한 선발 기준만 다음과 같이 다르다고 가정한다.

- N명 지원자의 이름, 1단계 점수, 면접 점수가 순서대로 입력된다.
- 이름은 공백을 포함하지 않는 최대 10자의 문자열이고, 문자열을 저장할 배열의 크기는 문자개수+1로 설정한다.
- 1단계 점수는 0~100점, 면접 점수는 0~100점의 정수로 입력된다.
- 최종 합격자의 이름, 1단계 점수 70%와 면접 점수 30%가 반영된 총점을 소수점 첫째 자리까 지 출력한다. 총점은 부동소수 자료형 double을 사용하시오.
- **최종 합격자의 이름과 총점뿐만 아니라, 1단계 점수, 면접 점수를 모두 출력**한다.
- **합격자 명단 아래에 최종 합격자의 수도 함께 출력**한다.
- **총점이 같은 동점자가 있으면, 1단계 점수가 더 높은 사람을 선발하고, 1단계 점수가 높은 사 람을 먼저 출력**한다.
- **M%명 이내의 합격자와 총점 및 1단계 점수가 모두 같은 동점자도 합격자로 선발하고, 입력한 순서가 빠른 사람을 출력**한다. (이름까지 같은 동점자는 없으며, 대문자 Z가 소문자 a보다 먼 저 나온다고 가정)
- **double형 자료의 대소 비교 시, 부동소수점으로 표현된 자료의 오차를 고려하여 비교**해야 한 다. (double형 자료 7*0.3과 3*0.7을 비교하면 같지 않음에 유의해서 비교할 것)

(힌트: 다음과 같이 계산하시오)

```
p[i].sum = p[i].firstStage * 7 + p[i].interview * 3;
printf("..... %.1fWn", p[i].name, ....., p[i].sum/10);
```

입력 예시 1

```
7 50      ↳ N=7, M=50, 3명
Kim 75 85
Lee 93 63
Park 94 82
Choi 78 70
Jung 65 85
Kang 83 90
Cho 90 70
```

출력 예시 1

```
Park 94 82 90.4
Kang 83 90 85.1
Lee 93 63 84.0 ↳ 총점이 같은 Lee, Cho 중에서
3              1단계 점수가 높은 Lee만 합격하고,
                Cho는 탈락
```

입력 예시 2

7 50 $\mapsto N=7, M=50, \underline{7*0.5}$ 명
 Kim 75 85
Lee 96 56
 Park 94 82
 Choi 78 70
 Jung 65 85
Kang 93 63
Cho 93 63

출력 예시 2

Park 94 82 90.4
 Lee 96 56 84.0
 Kang 93 63 84.0
 Cho 93 63 84.0
 4 \mapsto 총점이 같은 Lee, Kang, Cho 중에서
 1단계 점수가 높은 Lee부터 합격하고,
 총점 및 1단계 점수가 모두 같은 Kang, Cho
 는 모두 합격,
 총 합격자 수는 동점자를 포함한 4명

[문제 5] 공연 예약 정보를 출력하는 프로그램을 함수와 동적 할당을 이용하여 다음의 조건들을 지켜 작성하시오.

(1) 구조체 struct reserve와 struct tot_price를 다음과 같이 선언하여 사용한다.

<pre> struct reserve{ int id; // 고객 id char grade[4]; // 구매한 티켓의 등급 int price; // 구매한 티켓의 가격 }; </pre>	<pre> struct tot_price{ int id; // 고객 id int total_price; // 티켓 구매에 사용한 총 금액 }; </pre>
--	---

(2) 정수 N을 입력받고, 고객의 예매 정보를 저장할 크기 N의 struct reserve 타입의 배열을 동적으로 할당 받는다.

(3) N개의 예매 정보(고객 id와 구매한 티켓의 등급)를 입력받아, 티켓 가격과 함께 구조체 배열에 저장한다. 한 공연에는 4개 등급의 티켓이 있고, 각 티켓의 등급과 가격은 다음의 표와 같다.

티켓 등급	가격
VIP	140,000원
R	110,000원
S	85,000원
A	55,000원

(4) 예매 정보 입력을 위해 input 함수를 사용한다.

- 함수 원형: **void input(struct reserve *p, int N)**
- 구조체 배열의 시작 주소와 배열의 크기를 인자로 받아, 입력받은 고객 id, 티켓 등급과 티켓 가격을 저장한다.
- input 함수에서는 배열 표기 []를 사용하지 않고, 포인터를 움직이며 반복문을 구현한다.

(5) 예매를 한 고객의 수를 계산한다. 동일 고객이 여러 장의 티켓을 구매하는 경우, N개의 예매 정보에는 동일 고객의 id가 여러 번 나타난다. 중복된 id를 제외하여, **고객의 수를 계산한다**. 고객 수 계산에 cal_num 함수를 사용한다.

- 함수 원형: **int cal_num(struct reserve *p, int N)**
- 구조체 배열의 시작 주소와 배열의 크기를 인자로 받아, 고객의 수를 계산하여 반환한다.
- cal_num 함수에서는 배열 표기 []를 사용하지 않고, 포인터를 움직이며 반복문을 구현한다.

(6) **고객 수 크기의 struct tot_price 타입 배열을 동적으로 할당 받는다**.

(7) struct tot_price 타입의 구조체 배열에 고객 id와 해당 고객이 티켓 구매에 사용한 총 금액을 계산하여 저장한다. **동일 고객의 id가 반복 저장되어서는 안 된다**. 각 고객의 총 티켓 구매 금액 계산을 위해 cal_sum 함수를 호출하여 사용한다.

- 함수 원형: **void cal_sum(struct reserve *p, int N, struct tot_price *q)**
- 예매 정보가 저장된 구조체 배열의 시작 주소와 배열의 크기, 그리고, 고객의 총 티켓 구매 금액을 저장할 구조체 배열의 시작 주소를 인자로 받아, 고객 id를 저장하고, 해당 고객이 티켓 구매에 사용한 총 금액을 계산한다. **고객 id 저장 순서는 해당 고객 id가 예매 정보에 처음 나타나는 순이다**. (입출력예시 1 참고)
- cal_sum 함수에서는 배열 표기 []를 사용하지 않고, 포인터를 움직이며 반복문을 구현한다.

(8) struct tot_price 타입의 구조체 배열에 저장된 고객 id와 티켓 구매 총 금액을 main 함수에서 출력한다. (**출력시 id가 빠른 것부터 출력한다**). (**main 함수에서의 주소 표현 방식은 제한 사항 없다**).

입력 예시 1

```
10      ↳ N=10명
210 S
210 VIP
123 A
523 A
123 R
523 VIP
123 A
210 VIP
123 A
123 R
```

출력 예시 1

```
123 385000
210 365000
523 195000
↳ 세 명 고객의 id와 티켓 구매에 사용한 총 금액 출력
```

[문제 6] 문자열에 포함된 부분 문자열을 다른 문자열로 치환하는 프로그램을 문자열 함수를 이용하여 다음과 같이 작성 하시오.

- (1) 공백을 포함하는 문자열 하나를 입력받고, 공백을 포함하지 않는 두 개의 문자열을 추가로 입력받는다. 첫 번째 문자열을 문자열A, 두 번째와 세 번째 문자열을 각각 문자열B와 문자열C라 한다. 문자열A는 영문 대소문자, 공백, 그리고 구두점(.)들로 이루어져 있고, 문자열B와 문자열C는 영문 대소문자로만 이루어져 있다.
- (2) 문자열A에 포함된 모든 문자열B를 문자열C로 치환하고, 치환이 완료된 문자열을 문자열 출력(%s)으로 출력한다. 문자열A에 포함된 문자열B를 찾을 때, 대소문자를 구분한다. 즉, Abc와 abc는 같은 문자열이 아니다.
- (3) 문자열A의 최대길이는 100이고, 문자열B와 문자열C의 최대길이는 10이다. 문자열B와 문자열C의 길이는 같지 않을 수 있으며, 두 문자열 모두 문자열A 보다 짧다. 치환이 완료된 최종 문자열의 최대길이는 250이다.
- (4) 치환은 왼쪽 문자에서부터 순서대로 수행한다. (출력 예시 2 참조)
즉 AAA에는 AA가 1 개 있는 것으로 한다. AAAA에는 AA가 2개 있다. 즉 이미 한번 치환에 이용된 문자는 다음 치환에 사용하지 않는다.
- (5) 사용 가능한 문자열 함수: strcpy, strcat, strlen, strcmp, strncmp

(힌트) **strncmp** 함수 원형

```
int strncmp(char *lhs, char *rhs, int size)
```

(예) strncmp(st1, st2, N) 두 문자열에서 각 단어의 앞부분 N개 문자만 비교한다.

입력 예시 1

```
I have a pen. I have an applepen. PPAP.
pen      ↳ 문자열 B
mouse    ↳ 문자열 C
```

출력 예시 1

```
I have a mouse. I have an applemouse. PPAP.
```

입력 예시 2

출력 예시 2

Sequence GTCGTCG	↳ 문자열 A	Sequence CAGCTCG	↳ 왼쪽부터 순서대로 바뀜
GTCG	↳ 문자열 B		↳ 두 번째 GTCG는 바뀌지 않음
CAGC	↳ 문자열 C		

[문제 7] 문자열에 포함된 단어를 다른 단어로 치환하는 프로그램을 문자열 함수를 이용하여 다음과 같이 작성 하시오.

- (1) 공백을 포함하는 두 개의 문자열을 입력받는다. 첫 번째 문자열을 문자열A, 두 번째 문자열을 문자열B라 한다. 두 문자열 모두 영문 대소문자와 공백으로만 이루어져 있다.
- (2) **문자열A와 문자열B를 공백을 기준으로 단어로 나눈다.** (공백과 공백 사이의 문자열을 하나의 단어라 한다. 단어와 단어 사이에는 하나의 공백이 존재한다.)
- (3) **문자열A에 포함된 단어가 문자열A에 2회 이상 중복되어 나타나면 첫 번째 단어를 제외한 나머지 단어는 문자열B에 포함된 단어로 치환한다.** 치환이 완료된 문자열을 문자열 출력(%s)으로 출력한다.
- (4) 치환 작업은 문자열A의 첫 단어부터 순차적으로 진행되며, 문자열A의 각 단어에 대해 앞부분에 동일한 단어가 나온 적이 있는지 확인하고 치환한다. **치환될 단어는 문자열B의 첫 번째 단어부터 순차적으로 선택된다.** 단, 문자열B의 모든 단어는 문자열A에 포함되어있지 않다고 가정하고, 문자열B의 단어 수는 문자열A의 단어 수보다 많다고 가정한다.
- (5) 문자열A에 포함된 동일 단어는 대소문자를 구분한다. 즉, Abc와 abc는 같은 단어가 아니다.
- (6) 문자열A와 문자열B의 최대길이는 100이고, 치환이 완료된 최종 문자열의 최대길이는 250이다.
- (7) 사용 가능한 문자열 함수: strcpy, strcat, strlen, strcmp, strncmp

입력 예시 1

red orange red yellow green red blue purple yellow	↳ 문자열 A
white black gray pink brown blush crimson garnet vermillion indigo	↳ 문자열 B

출력 예시 1

red orange white yellow green black blue purple gray
↳ 문자열A의 중복 단어 red/red/yellow는 문자열B의 첫 번째, 두 번째, 세 번째 단어인 white/black/gray로 치환