

<고급C 프로그래밍 및 실습> 과제 - 2차 예상문제 문제지

(주의) 2차 인증 날짜 2022.11.21.(월)

※ 문제지의 무단 배포 및 사용을 원칙적으로 금지합니다.

[문제 1-1] (50점) 공백을 포함하는 알파벳 소문자로 된 하나의 문장을 입력받고, 이를 단어 단위로 분리한 후 이를 다시 영어 사전순으로 정렬하시오. 과제2 [문제 2-1]

- 문장을 단어 단위로 분리하여 '이차원 배열'에 저장한다.
- 문장을 입력받기 위해 gets 함수 사용한다. 문장의 길이는 최대 200 이다.
- 표준 문자열 처리함수 사용 가능하다. (예: strlen, strcpy, strcmp 등)
- 단어의 개수는 최대 20개이고, 각 단어의 최대길이는 (널 문자 포함하여) 200 이다.
- 문자열 입력 시 gets 함수를 사용한다. (위반 시 50% 감점)
- 단어 단위로 분리하기 위해 strtok 함수를 사용하지 않는다. (위반 시 50% 감점)
- 문자열 정렬을 위한 문자열 비교 시 strcmp를 이용한다. (위반 시 각 50% 감점)

입력 예시 1

simple is best

출력 예시 1

simple ↳ 단어 분리하여 출력하기
is
best
best is simple ↳ 분리된 단어 정렬하여 출력하기

입력 예시 2

i am a boy

출력 예시 2

i ↳ 단어 분리하여 출력하기
am
a
boy
a am boy i ↳ 분리된 단어 정렬하여 출력하기

[문제 2-1] (50점) 5명 학생의 이름과 각 학생의 학번 및 지난학기 평균학점을 입력받는다. 그다음 찾고자 하는 학생의 이름을 입력하면, 이 학생의 학번과 지난학기 평균학점을 출력하시오.

과제3 [문제 1-1]

- 한 학생의 정보는 다음과 같다. (struct student)
 - 이름 : 공백을 포함하지 않고 길이가 최대 9인 문자열 (널 문자까지 10 바이트만 선언하기)
 - 학번 : 공백을 포함하지 않고 길이가 10인 문자열이고, 첫 4자리는 입학년도를 의미
 - 지난학기 평균학점 : A, B, F 중 한 개의 학점 (문자)
- 학생들의 이름은 모두 다르고, 대소문자를 구분한다. (예: Lee와 lee는 다른 학생이다)
- 학생들의 입학년도는 2017년에서 2019년 사이로 가정한다.

입력 예시1

```
lee1 2017102656 A
lee2 2018111823 A
park 2018059961 A
choi 2018010101 F
lee3 2019565656 B
park
```

출력 예시1

```
□ 2018059961 A
```

[문제 3-1] (30점) K 전자는 필기시험 점수의 80%와 면접시험 점수의 20%를 합산한 총점이 높은 순으로, N명의 지원자 중 M명의 신입직원을 선발 한다. 단, **총점이 같은 동점자는 없다고 가정한다.** 각 지원자의 이름과 필기시험, 면접시험 점수가 아래 예시와 같이 주어질 때 M명 합격자의 이름과 최종 점수를 출력하는 프로그램을 작성하시오. **1차 모의 [문제 5-1]**

- 먼저, 지원자 수 N과 합격자 수 M이 입력된다. $N \leq 100$ 이고, $M \leq N$ 이다.
- 다음은 N명 지원자의 이름, 필기시험 점수, 면접시험 점수가 순서대로 입력된다.
- 이름은 공백을 포함하지 않는 최대 10자의 문자열이며, 필기시험과 면접시험 점수는 100점 만점이다.
- 합격자의 이름, 필기시험 점수의 80%와 면접시험 점수의 20%가 반영된 총점을 소수점 첫째 자리까지 출력한다. 부동소수 자료형은 double을 사용하시오.
- **출력순서**는 선발순서(총점이 높은 합격자부터 낮은 합격자 순) 이다.
- main() 함수에서 지원자를 정렬하는 과정에 아래의 swap() 함수를 호출하여 사용한다.

(1) swap() 함수

- 함수 원형: `void swap(struct person *p, struct person *q)`
- p와 q는 구조체 배열의 두 원소의 주소이며, 두 원소의 내용을 바꾸는 작업을 한다.

▶ 구조체 배열을 사용하지 않은 경우 (100% 감점)

▶ swap() 함수를 사용하지 않거나 함수의 원형을 지키지 않은 경우 (20% 감점)

▶ swap() 함수 내에서 구조체 대입 연산을 사용하지 않은 경우 (10% 감점)

입력 예시 1

```
7 3      ↳ N=7, M=3
Aaaaaa 75 85
Bbbbbb 86 77
Ccccc 94 82
Ddddd 78 70
Eeeee 65 85
Fffff 83 90
Ggggg 90 70
```

출력 예시 1

```
Ccccc 91.6
Ggggg 86.0
Fffff 84.4
```

[문제 4-1] (50점) 종료조건을 만족할 때까지 (종료조건 -1을 포함하여 최대 100개의) 한 자리 양의 정수를 계속 입력 받아 배열 x에 저장하고, 정수 M을 입력받아, x[M]부터 값이 **연속적으로 증가하는 원소들만** 모아 하나의 정수를 만들어 출력하는 프로그램을 아래 함수들을 사용하여 작

성 하시오. 예를 들어, x[M]부터 입력된 정수가 2, 4, 6, 7, 4, 3, -1인 경우, 값이 증가하는 5까지의 정수들인 2, 4, 6, 7을 모아 만든 정수 2467를 출력한다. 입력 종료조건은 -1의 입력이다.

1차 모의 2-1

(1) input 함수

- 함수 원형: `int input(int *p)`
- 배열 x의 시작 주소를 인자로 받아 종료 조건까지 정수를 입력받아 배열에 저장하고, 배열 원소의 개수를 반환한다.

(2) sel_next 함수

- 함수 원형: `int *sel_next(int *p, int N, int M)`
- 배열 x의 시작 주소와 배열의 크기 N, 원소 값의 연속적 증가 여부 검사를 시작할 원소의 위치 M을 인자로 받아, 원소 값의 증가가 끝나는 마지막 원소의 주소를 반환 한다. 만약, x[M] 다음 원소 값이 x[M]의 값 보다 작거나 같은 경우에는 x[M]의 주소를 반환한다.

(3) number 함수

- 함수 원형: `int number(int *p, int *q)`
- x[M]의 위치와 sel_next에서 반환된 위치를 인자로 받아, 두 포인터 사이의 한 자리 정수를 모아, 하나의 정수로 만들어 반환 한다.

▶ 배열의 변수 선언 이외의 배열 표기 []는 사용 금지 (포인터 표기 사용) (**위반 시 100% 감점**)

▶ 배열을 반복문으로 훑어볼 시, 주소를 이용하여 포인터를 이동시키며, 반복문 구현 (**위반 시 100% 감점**)

▶ 전역변수 사용 금지 (**위반 시 100% 감점**)

▶ input, sel_next, number 함수를 사용하지 않거나, 함수의 원형을 지키지 않거나, 함수를 목적에 맞게 사용하지 않은 경우 감점 (**input, number 함수는 각 25%, sel_next 함수는 50% 감점**)

▶ 정수를 만들어 출력하지 않고, 배열의 각 원소를 연속으로 출력하는 경우 감점 (**100% 감점**) 즉 number 함수를 제대로 구현하여 main 함수에서 정수를 출력하면 감점 없음.

입력 예시 1

1 2 4 6 7 4 3 -1
1

↳ M=1

출력 예시 1

2467

↳ x[1]인 2부터 연속적으로 증가하는 7까지 모아 2467 출력

입력 예시 2

1 2 3 4 5 4 3 -1
4

↳ M=4

출력 예시 2

5

↳ x[4]인 5 이후 증가하는 수가 없으므로 5 출력

[문제 4-2] (100점) 종료조건을 만족할 때까지 (종료조건 -1을 포함하여 최대 100개의) 한 자리 양의 정수를 입력 받아 배열 x에 저장한 후, 배열의 원소가 연속적으로 증가하는 구간과 연속적으로 감소하는 구간으로 나누어, 연속적으로 증가하는 구간의 원소들을 모아 하나의 정수로 만들고, 연속적으로 감소하는 구간의 원소들을 모아 하나의 정수로 만들어 출력하는 프로그램을 아래 함수들을 사용하여 작성 하시오. 예를 들어, 사용자로부터 입력된 정수가 다음과 같은 경우

1	2	3	4	5	4	3	2	3	4	2	-1
---	---	---	---	---	---	---	---	---	---	---	----

첫 번째 원소인 1부터 값이 증가하는 5까지의 정수를 모아, 12345를 출력하고, 5부터 값이 감소하는 2까지 모아 5432를 출력하고, 다시 값이 증가하는 2, 3, 4를 모아 234를 마지막으로 값이 감소하는 구간인 4와 2를 모아 42를 출력한다. 숫자가 증가하다가 감소하거나, 감소하다가 증가하는 지점의 숫자는 만들어지는 두 정수에 모두 포함된다. 연속해서 동일한 숫자가 입력되는 경우는 없다고 가정하며, 입력 종료조건은 -1의 입력이다. 예시 2와 같이 첫 번째 원소부터 감소하다가 증가할 수도 있다.

- (1) input 함수, [문제 2-1]과 동일
 - 함수 원형: `int input(int *p)`
 - 배열 x의 시작 주소를 인자로 받아 종료 조건까지 정수를 입력받아 배열에 저장하고, 배열 원소의 개수를 반환한다.
- (2) sel_next 함수, [문제 2-1]과 다름
 - 함수 원형: `int *sel_next(int *p)`
 - 배열의 한 원소의 주소 p를 인자로 받아, p가 가리키는 원소부터 원소 값의 **증가 또는 감소**가 끝나는 마지막 원소의 주소를 반환 한다. 위 예제의 경우, x[4]의 주소가 인자로 전달되면, x[7]의 주소가 반환된다.
- (3) number 함수, [문제 2-1]과 동일
 - 함수 원형: `int number(int *p, int *q)`
 - 배열의 두 원소의 주소 p와 q를 인자로 받아, 두 포인터 사이의 한 자리 정수를 모아 하나의 정수로 만들어 반환 한다.

- ▶ 배열의 변수 선언 이외의 배열 표기 []는 사용 금지 (포인터 표기 사용) (위반 시 100% 감점)
- ▶ 배열을 반복문으로 훑어볼 시, 주소를 이용하여 포인터를 이동시키며, 반복문 구현 (위반 시 100% 감점)
- ▶ 전역변수 사용 금지 (위반 시 100% 감점)
- ▶ input, sel_next, number 함수를 사용하지 않거나, 함수의 원형을 지키지 않거나, 함수를 목적에 맞게 사용하지 않은 경우 감점 (input, number 함수는 각 25%, sel_next 함수는 50% 감점)
- ▶ 정수를 만들어 출력하지 않고, 배열의 각 원소를 연속으로 출력하는 경우 감점 (100% 감점) 즉 number 함수를 제대로 구현하여 main 함수에서 정수를 출력하면 감점 없음.

입력 예시 1

출력 예시 1

1 2 3 4 5 4 3 2 3 4 2 -1	12345 5432 234 42
--------------------------	----------------------------

입력 예시 2

8 6 4 2 1 0 3 5 7 1 4 -1

출력 예시 2

864210

357 ↳ 0은 두 번째 정수에 포함이 되지만, 0357이 아니라 357 출력

71

14

[문제 5-1] (50점) 문자열에 포함된 부분 문자열을 다른 문자열로 치환하는 프로그램을 문자열 함수를 이용하여 다음과 같이 작성 하시오. **1차 모의 4-1**

- (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.

- 프로그램 작성 시 다음의 조건들이 지켜져야 한다.

- ▶ **strcpy, strcat, strlen, strcmp, strncmp 이외의 다른 문자열 함수 사용 시 (50% 감점)**
- ▶ **문자열을 치환하여 하나의 문자열을 만들어 문자열 출력(%s)하지 않고, 문자 출력(%c)을 하거나, 문자열 출력(%s)을 반복해서 하는 경우 (100%감점)**

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

Sequence GTCGTCG	↳ 문자열 A
GTCG	↳ 문자열 B
CAGC	↳ 문자열 C

출력 예시 2

Sequence CAGCTCG	↳ 왼쪽부터 순서대로 바뀜
	↳ 두 번째 GTCG는 바뀌지 않음

제출기한 및 방법

- * OJ시스템(<https://ex-oj.sejong.ac.kr/index.php/auth/login>) 내의 2차 인증 예상문제를 이용하여 제출
- * 제출 마감: **11월 11일(금요일)** 밤 11시 59분까지 제출
- * 과제 점수는 위 마감일 전에 제출된 가장 마지막 코드를 기준으로 부여
(마감일 이후에 제출되는 코드는 채점에서 제외됨)
- * OJ시스템의 점수는 참고로만 사용
 - 문제의 조건을 만족시켜서 코딩했는지 조사 후 최종 점수 부여
 - 코드 2줄당 1줄의 비율로 반드시 주석을 적을 것(주석이 없을 경우 감점)
 - 보고서 등 기타 제출물 없음
 - 문제5는 심화반 문제로 채점에 반영하지 않으나 참고용으로 풀이 권장
 - 카피본 발견 시 원본/카피본 모두 0점 처리
 - 모든 문제들이 이미 해답을 제공한 문제들입니다. 본인이 직접 해결할 수 있는 실력을 쌓아서 2차 인증에 대비하기 바랍니다.