

〈C프로그래밍 및 실습〉 4차 모의고사 문제지

[문제 1] (100점) 종료 조건 시까지 문자를 입력받아, 영어 대문자는 소문자로, 소문자는 대문자로 변환하여 배열에 저장한다. 단, 영어 대문자와 소문자 이외의 문자는 배열에 저장하지 않는다. 배열에 저장된 문자를 출력 한 후, 배열에 저장된 문자 중, (대문자 자음 + 소문자 모음 + 소문자 자음)이 연속으로 나타나는 횟수를 출력하는 프로그램을 작성하시오. 단, 영문자가 포함되지 않은 문자열이 입력되는 경우는 없다고 가정한다.

- 종료 조건: 문자 '*' 입력. 입력 문자는 최대 100개 임.
- 대문자 모음은 'A', 'E', 'I', 'O', 'U'이고, 소문자 모음은 'a', 'e', 'i', 'o', 'u' 임.

입력 예시 1

출력 예시 1

123c#456A!789T\$*	Cat 1	↳ 영문자 이외의 문자는 제외 ↳ Cat 1개
-------------------	----------	------------------------------

입력 예시 2

출력 예시 2

bOBj#OBXY!ab333gOODdAYfOX777*	BobJobxyABGoodDayFox 4	↳ 영문자 이외의 문자는 제외 ↳ Bob, Job, Day, Fox 4개
-------------------------------	---------------------------	---

※ [문제 2-1]에서 [문제 2-3]까지는 연관된 문제이며, 배점이 각각 40, 70, 100점이다. 가장 높은 점수 하나만 반영하며, 합산하지 않는다.

[문제 2-1] (40점) 양의 정수 N을 입력받고, N의 각 자릿수의 숫자를 거꾸로 하여 만든 정수를 출력하는 프로그램을 reverse_num() 함수를 사용하여 작성하시오. main함수에서는 양의 정수 하나를 입력받고, reverse_num() 함수를 호출하여, 거꾸로 된 정수를 반환받아 출력한다. 정수 N은 최대 9자리 수라고 가정한다. 함수원형선언을 하여, reverse_num() 함수의 정의는 main함수 뒤에 오도록 작성 한다.

- reverse_num 함수
 - 인자 : int형 변수 1개
 - 정수의 각 자릿수의 숫자를 거꾸로 하여 만든 정수를 반환
 - 반환값 : int형

- ▶ 인자와 반환값의 자료형과 인자 수가 맞지 않은 경우 (100% 감점)
- ▶ reverse_num 함수를 사용하지 않거나, 함수 호출을 잘못된 경우 (100% 감점)
- ▶ 함수원형선언을 하지 않거나 잘못된 경우, 또는 reverse_num 함수 본체를 main 함수 전에 작성한 경우 (20% 감점)
- ▶ reverse_num 함수 내에서 출력을 하거나, 숫자를 하나씩 출력한 경우 (100% 감점)
- ▶ 전역변수를 사용한 경우 (100% 감점)

입력 예시 1

출력 예시 1

34830332	23303843
----------	----------

입력 예시 2

출력 예시 2

912175190	91571219	↳ 제일 첫 0은 출력되지 않음
-----------	----------	-------------------

[문제 2-2] (70점) 정수 N을 입력받고, N의 각 자릿수의 숫자 중 중복된 숫자가 있으면, 나중에 나오는 숫자를 삭제하여 중복된 숫자가 없는 정수를 만들어 출력하는 프로그램을 del_dupnum() 함수를 사용하여 작성하시오. main함수에서는 양의 정수 하나를 입력받고, del_dupnum() 함수를 호출하여, 중복된 숫자가 없는 수를 반환 받아 출력한다. 정수 N은 최대 9자리 수라고 가정한다. 함수원형선언을 하여, del_dupnum() 함수의 정의는 main함수 뒤에 오도록 작성 한다.

- del_dupnum 함수

- 인자 : int형 변수 1개
- 정수의 각 자릿수의 숫자 중 나중에 나온 중복된 숫자를 제외하여 만든 정수 반환
- 반환값 : int형

- ▶ 인자와 반환값의 자료형과 인자 수가 맞지 않은 경우 (100% 감점)
- ▶ del_dupnum 함수를 사용하지 않거나, 함수 호출을 잘못된 경우 (100% 감점)
- ▶ 함수원형선언을 하지 않거나 잘못된 경우, 또는 del_dupnum 함수 본체를 main 함수 전에 작성한 경우 (20% 감점)
- ▶ del_dupnum 함수 내에서 출력을 하거나, 숫자를 하나씩 출력한 경우 (100% 감점)
- ▶ 전역변수를 사용한 경우 (100% 감점)

입력 예시 1

출력 예시 1

23303843	23084	↳ 첫 번째 3은 두고 뒤에 나온 3은 모두 제거
----------	-------	-----------------------------

입력 예시 2

출력 예시 2

891571219	891572	↳ 뒤에 나온 1, 9 제거
-----------	--------	-----------------

[문제 2-3] (100점) 종료 조건 시까지 정수 N을 반복해서 입력받아, 아래의 작업을 순차적으로 수행하는 프로그램을 작성하시오. 정수 N은 최대 9자리 수라고 가정한다.

- 종료 조건: 음수 입력 (입력된 음수에 대해서는 아래 작업을 수행하지 않음)

- 1) [문제 2-1]의 reverse_num()함수를 이용하여 정수 N을 거꾸로 변경한다. (출력은 안함)
- 2) 1단계 결과로 만들어진 정수에 대해, [문제 2-2]의 del_dupnum()함수를 이용하여 중복된 숫자들을 제외한 정수 M을 만들어 main함수에서 출력한다.
- 3) 입력된 정수 N과 1, 2단계를 거쳐 만들어진 정수 M의 차이가 가장 큰 수를 찾아 입력 정수 N과 함께 그 차이를 main함수에서 출력한다. N과 M의 차이는 N-M의 절대값으로 계산한다. 만일 차이가 같은 경우가 있으면 제일 먼저 입력된 정수와 차이를 출력한다.
- 4) 함수원형선언을 하여, reverse_num() 함수와 del_dupnum() 함수의 정의는 main함수 뒤에 오도록 작성 한다.

- ▶ 인자와 반환값의 자료형과 인자 수가 맞지 않은 경우 (100% 감점)
- ▶ reverse_num, del_dupnum 함수를 사용하지 않거나, 함수 호출을 잘못된 경우 (100% 감점)
- ▶ 함수원형선언을 하지 않거나 잘못된 경우, 또는 reverse_num, del_dupnum 함수 본체를 main 함수 전에 작성한 경우 (20% 감점)

- ▶ del_dupnum 함수 내에서 출력을 하거나, 숫자를 하나씩 출력한 경우 (100% 감점)
- ▶ 전역변수를 사용한 경우 (100% 감점)

입력 예시 1	출력 예시 1
34830332 912175 1983 101224 1357 -5	□23084□57129□3891□4210□7531 34830332 34807248 ↳ 34830332 → 23303843 → 23084 (입력 N) → (N을 거꾸로한 수) → (중복 제거 결과 M) ↳ 34830332 - 23084 는 34807248 (차이가 가장 큼)
입력 예시 2	출력 예시 2
12345 54231 -9	□54321□13245 12345 41976 ↳ 54231 → 13245 → 13245 (입력 N) → (N을 거꾸로한 수) → (중복 제거 결과 M) ↳ 12345-54321 =41976, 54231-13245 =40986, 차이가 큰 값은 41976

※ [문제 3-1]에서 [문제 3-2]까지 연관된 문제이며, 배점이 각각 50, 100점이다. [문제 3-1]을 안 풀고 [문제 3-2]만 풀어도 100점을 부여한다. 가장 높은 배점 하나만 반영한다. 합산하지 않는다.

※ [문제 3-1]과 [문제 3-2]는 다음의 조건에 맞춰 프로그램이 작성되어야 한다. 조건을 맞추지 못한 경우 아래와 같이 감점이 부여된다.

- ▶ 각 문제에서 지정한 함수를 사용하지 않거나, 함수를 의도에 맞게 사용하지 않은 경우 (함수마다 각 50% 감점)
- ▶ 지정된 함수의 인자와 반환값의 자료형 또는 인자의 수가 맞지 않은 경우 (함수마다 각 50% 감점)
- ▶ 함수원형선언을 하지 않거나 선언을 잘못된 경우, 또는 지정된 함수를 main 함수 전에 작성한 경우 (함수마다 각 10% 감점)
- ▶ 전역변수를 사용한 경우 (100% 감점)

[문제 3-1] (50점) 종료 조건(0 또는 음수 입력)까지 정수 N을 반복해서 입력받아, 아래의 두 조건 중 한 조건이라도 만족하는 정수를 배열 X에 저장한다. 배열 X의 크기는 100이다.

- (1) N이 소수(prime number)인 경우
- (2) N의 각 자릿수가 홀수만으로 이루어진 경우

배열 X에 저장된 원소를 가장 마지막 원소부터 역순으로 출력한다. 만약, 입력된 정수 중 (1) 또는 (2)의 조건을 만족하는 정수가 하나도 없는 경우에는 0을 출력한다.

조건 (1)과 (2)를 판단하기 위해 prime_number() 함수와 odd_number() 함수를 사용한다.

- prime_number 함수
 - 함수 원형: `int prime_number(int num)`
 - 양의 정수 num이 소수이면 1을 반환하고, 소수가 아니면 0을 반환한다.
- odd_number 함수
 - 함수 원형: `int odd_number(int num)`

- 양의 정수 num의 각 자릿수가 모두 홀수이면 1을 반환하고, num의 각 자릿수 중 짝수가 포함되어 있으면 0을 반환한다.

입력 예시 1

1493 755 719 850 883 12580 11953 95971 139 1455 -1

출력 예시 1

139 95971 11953 883 719 755 1493

↳ 1493, 719, 883, 11953, 95971, 139는 소수라서 출력, 755는 소수는 아니지만 모든 자릿수가 7, 5, 5로 홀수라 출력, 출력 순서는 배열에 저장된 역순

입력 예시 2

25 45 65 85 105 125 145 165 185 205 -1

출력 예시 2

0

↳ 소수도 없고, 모든 자릿수가 홀수인 수도 없으므로 0 출력

[문제 3-2] (100점) 종료 조건(0 또는 음수 입력)까지 정수 N을 반복해서 입력받아,

아래의 두 조건 중 한 조건이라도 만족하는 정수를 배열 X에 저장한다. 배열 X의 크기는 100 이다.

- (1) N이 소수(prime number)인 경우
- (2) N의 각 자릿수가 홀수만으로 이루어진 경우

조건 (1)과 (2)를 판단하기 위해 **prime_number()** 함수와 **odd_number()** 함수를 사용한다.

배열 X에 저장된 수 중, 조건 (1)과 (2)를 동시에 만족하는 수가 **연속적으로 나온 최대 횟수**를 카운트하고, 해당 구간의 조건 (1)과 (2)를 동시에 만족하는 수를 모두 출력하는 프로그램을 **count_number()** 함수를 이용하여 작성하시오.

만약, 입력된 정수 중 조건 (1)과 (2)를 동시에 만족하는 수가 하나도 없으면 0을 출력한다.

- prime_number 함수

- 함수 원형: **int prime_number(int num)**
- 양의 정수 num이 소수이면 1을 반환하고, 소수가 아니면 0을 반환한다.

- odd_number 함수

- 함수 원형: **int odd_number(int num)**
- 양의 정수 num의 각 자릿수가 모두 홀수이면 1을 반환하고, num의 각 자릿수 중 짝수가 포함되어 있으면 0을 반환한다.

- count_number 함수

- 함수 원형: **int count_number(int cnt, int num)**
- (1)과 (2)의 조건을 모두 만족하는 연속된 수를 세는 cnt 값과 양의 정수 num을 전달받아, num이 두 조건을 모두 만족하면, cnt+1한 값을 반환하고, 그렇지 않으면, 0을 반환한다.
- num 값이 두 조건을 모두 만족하는지 확인하기 위해 count_number() 함수 내에서 prime_number() 함수와 odd_number() 함수를 호출한다.

- **최대 길이가 같은 구간이 있으면 제일 처음 나온 구간의 정보를 출력한다.**

입력 예시 1

1493 755 **719** 850 883 12580 **11953 95971 139** 1455 -1

출력 예시 1

11953□95971□139□

↳ 소수이면서 각 자릿수가 홀수로만 이루어진 수가 3번 연속 나온 구간이 제일 길어, 해당 구간의 수를 **배열에 저장된 순서대로 출력**

입력 예시 2

1493 755 850 883 12580 1455 -1

출력 예시 2

0

↳ 소수이면서 각 자릿수가 홀수로만 이루어진 수가 입력되지 않아 0 출력

※ [문제 4-1]에서 [문제 4-2]까지 연관된 문제이며, 배점이 각각 50점, 100점이다. [문제 4-1]을 안 풀고 [문제 4-2]만 풀어도 100점을 부여한다. 가장 높은 배점 하나만 반영하며, 합산하지 않는다.

[문제 4-1] (50점) 회문수(palindromic number)는 순서대로 읽은 수와 거꾸로 읽은 수가 일치하는 수를 말한다. 즉, 회문수는 대칭인 수를 의미하며, 예를 들어 34543은 회문수이고, 34567은 회문수가 아니다.

(1) 양의 정수 N을 입력받는다. N은 최대 9자리 정수이다.

(2) reverse_num 함수를 호출하여, N의 각 자릿수를 역순으로 하여 만든 정수를 반환받고, is_equal 함수를 호출하여, N이 회문수인지 판단한다.

(3) 정수 N과 N을 거꾸로 읽은 수 M, 그리고 N이 회문수 인지 여부를 나타내는 1 또는 0(N이 회문수이면 1, 회문수가 아니면 0)을 main 함수에서 화면에 출력한다.

(4) main 함수 위에 함수원형선언을 하고, 함수의 정의는 main함수 아래에 오도록 작성 한다.

- reverse_num 함수

○ 함수 원형 : **int reverse_num(int N)**

○ 인자 : 정수 N

○ 반환 값 : 정수 N의 각 자릿수의 숫자를 역순으로 하여 만든 정수를 반환

- is_equal 함수

○ 함수 원형 : **int is_equal(int N, int M)**

○ 인자 : 정수 N과 M

○ 반환값 : N과 M이 같으면 1을, 같지 않으면 0을 반환

○ is_equal 함수는 정수 N과 N을 거꾸로 읽은 수 M이 같은지 판단하기 위해 사용한다.

▶ 함수를 설명과 다르게 정의하거나, 목적에 맞지 않게 호출하여 사용한 경우 (각 함수 50%씩 감점)

▶ 함수원형선언을 올바르게 하지 않거나, 함수를 main 함수 전에 작성한 경우 (각 함수 10%씩 감점)

▶ 전역변수를 사용한 경우 (50% 감점)

입력 예시 1	출력 예시 1
12321	12321 12321 1 ↳ 회문수이므로 세 번째 숫자는 1을 출력
입력 예시 2	출력 예시 2
4566540	4566540 456654 0 ↳ 거꾸로 읽은 수 앞에 불필요한 0은 출력되지 않음 ↳ 회문수가 아니므로 세 번째 숫자는 0을 출력

[문제 4-2] (100점) 회문수(palindromic number)는 순서대로 읽은 수와 거꾸로 읽은 수가 일치하는 수를 말한다. 즉, 회문수는 대칭인 수를 의미하며, 예를 들어 34543은 회문수이고, 34567은 회문수가 아니다.

(1) 양의 정수 N과 삭제할 자릿수를 나타내는 정수 T를 입력받는다. N은 최대 9자리 정수이고, T는 (N의 전체 자릿수)/2 보다 작거나 같다. 즉, N이 7자리 정수일 경우, T는 1, 2 또는 3이다.

(2) `reverse_num` 함수를 호출하여, N의 각 자릿수를 역순으로 하여 만든 정수를 반환받고, `is_palindromic` 함수를 호출하여, N이 회문수인지 판단한다.

(3) `del_digit` 함수를 호출하여, N이 회문수이면, N에서 오른쪽으로부터 T번째 자릿수와 T와 대칭되는 곳에 위치한 자릿수를 삭제한 정수를, N이 회문수가 아니면, N에서 오른쪽으로부터 T번째 자릿수만 삭제된 정수를 반환받는다. 여기서 자릿수는 정수의 오른쪽부터 센다. 즉, 일의 자리가 1번째 자리, 십의 자리가 2번째 자리, ... 이다.

예시 1) N=34543, T=1이면, 1번째와 5번째 자리의 숫자 3을 모두 삭제한 454를 반환

예시 2) N=345543, T=3이면, 3번째와 4번째 자리의 숫자 5를 모두 삭제한 3443을 반환

예시 3) N=34567, T=1이면, 1번째 자리의 숫자 7을 삭제한 3456을 반환

(4) 정수 N과 N을 거꾸로 읽은 수 M, 그리고 N이 회문수 인지 여부를 나타내는 1 또는 0(N이 회문수이면 1, 회문수가 아니면 0)과 N에서 T번째 자릿수(N이 회문수이면, T와 대칭되는 곳의 자릿수도 포함)를 삭제한 수를 차례로 `main` 함수에서 화면에 출력한다.

(5) `main` 함수 위에 함수원형선언을 하고, 함수의 정의는 `main`함수 아래에 오도록 작성 한다.

- `reverse_num` 함수

◦ 함수 원형 : `int reverse_num(int N)`

◦ 인자 : 정수 N

◦ 반환 값 : 정수 N의 각 자릿수의 숫자를 역순으로 하여 만든 정수를 반환

- `is_palindromic` 함수

◦ 함수 원형 : `int is_palindromic(int N)`

◦ 인자 : 정수 N

◦ 반환값 : N이 회문수이면 1을, 그렇지 않으면 0을 반환

◦ N의 회문 여부를 판단하기 위해, `reverse_num` 함수를 호출하여 사용한다.

- `del_digit` 함수

◦ 함수 원형 : `int del_digit(int N, int T)`

◦ 인자 : 정수 N과 T

◦ 반환값 : N이 회문수이면, N의 T번째 자릿수와 T번째 자리와 대칭되는 자릿수를 삭제한 수를, N이 회문수가 아니면, N의 T번째 자릿수만 삭제한 수를 반환한다.

◦ N의 회문 여부를 판단하기 위해 `is_palindromic` 함수를 호출하여 사용한다.

▶ 함수를 설명과 다르게 정의하거나, 목적에 맞지 않게 호출하여 사용한 경우 (함수 세 개 차례대로 각 25%, 25%, 50% 감점)

- ▶ 함수원형선언을 올바르게 하지 않거나, 함수를 main 함수 전에 작성한 경우 (각 함수 10%씩 감점)
- ▶ 전역변수를 사용한 경우 (50% 감점)

입력 예시 1

출력 예시 1

12345 2	12345 54321 0 1235 ↳ 회문수가 아니므로 2번째 자리를 삭제한 1235를 출력
---------	--

입력 예시 2

출력 예시 2

78987 1	78987 78987 1 898 ↳ 회문수이므로 1번째, 그리고 대칭된 5번째 자리를 삭제한 898을 출력
---------	--

입력 예시 3

출력 예시 3

99 1	99 99 1 0 ↳ 회문수이므로 1번째, 그리고 대칭된 2번째 자리를 삭제한 0을 출력
------	--