

## <C프로그래밍 및 실습> 8장 함수 실습 문제

### ※ 문제에 대한 안내

- 출력 예시에서 □는 공백을 의미한다.
- 입출력 예시에서  $\mapsto$  이 후는 각 입력과 출력에 대한 설명이다.

**8장3절 [ 문제 1 ]** 양의 정수 N을 입력 받아, 1부터 N까지의 합을 출력하는 프로그램을 작성하시오.

- add 함수 정의 및 사용
  - 인자는 두 개의 정수 x와 y, 반환형은 int
  - x와 y의 합을 반환
- main 함수
  - N값을 입력 받고, add 함수를 반복 호출하여 N까지의 합 계산
  - main 함수에서는 더하기 연산자(+) 사용 금지. 단, 증감연산자(++ or --)는 사용 가능
- 힌트: 기존의 `sum = sum + i ;` 대신에, `sum = add( __ , __ ) ;`

입력 예시

출력 예시

10	55
----	----

**8장3절 [ 문제 2 ]** 두 개의 양의 정수 A과 B를 입력 받아, A부터 B까지의 합을 계산하는 프로그램을 작성하시오.

- $A \leq B$ 인 입력만 들어온다고 가정하시오.
- sum 함수 정의 및 사용
  - 인자는 하나의 정수 n, 반환형은 int
  - 1부터 n까지의 합 반환,  $n(n+1)/2$  공식 이용

입력 예시

출력 예시

4 8	30
-----	----

**8장3절 [ 문제 3 ]** 하나의 양의 정수 N을 입력 받아 다음 수식의 결과를 출력하는 프로그램을 작성하시오.

$$1 + (1+2) + (1+2+3) + (1+2+3+4) + \dots + (1+2+\dots+N)$$

- sum 함수 정의 및 사용
  - 문제 2와 동일

입력 예시

출력 예시

4	20
---	----

**8장4절 [ 문제 13 ]** 하나의 양의 정수 N을 입력 받아, 1부터 N까지의 정수에서 숫자 3이 총 몇 번 나타나는지를 출력하는 프로그램 작성하시오.

- N=33 일 때, 3, 13, 23, 30, 31, 32, 33, 총 7개의 수에서 8개의 3이 나타난다.  
(∵ 숫자 33에는 3이 두 개 있으므로)
- **count\_3 함수** 정의 및 사용
  - 인자는 하나의 정수 x, 반환형은 int
  - 하나의 정수 x에 숫자 3이 나타나는 횟수 반환
- 함수원형을 선언하고, 함수 정의를 main 함수 뒤에 작성하시오.

입력 예시 1

출력 예시 1

33	8
----	---

입력 예시 2

출력 예시 2

31323	13711
-------	-------

**8장4절 [ 문제 14 ]** 두 양의 정수 N과 M을 입력 받아, N보다 큰 소수를 작은 것부터 차례로 M 개 출력하는 프로그램을 작성하시오.

- is\_prime 함수
  - 인자는 하나의 정수 x, 반환형은 int
  - 정수 x가 소수이면 1 반환, 소수가 아니면 0 반환
- next\_prime 함수
  - 인자는 하나의 정수 x, 반환형은 int
  - 정수 x 보다 크면서 가장 작은 소수를 반환 (is\_prime 함수 사용)
- main 함수
  - next\_prime 함수를 **반복 호출하여**, M개의 소수를 출력
- 함수원형을 선언하고, 함수 정의를 main 함수 뒤에 작성하시오.

입력 예시

출력 예시

31 5	□37 41 43 47 53
------	-----------------

## 11주차 실습문제

[ 문제 1 ] N개의 영문자를 입력받아, 모두 대문자로 변환하여 출력하는 프로그램을 작성하라.

- 소문자를 대문자로 변환하기 위해 `atoA` 함수를 정의하여 사용한다. (단, *영문자가 아니면 그대로 출력*, N은 최대 20)

- **atoA** 함수 정의 및 사용
  - 인자는 한 개의 문자
  - 이 함수는 한 개의 문자를 대문자로 변환 후 반환
- **main** 함수
  - 10개의 문자를 입력 받고, **atoA** 함수를 호출하여 소문자를 대문자로 변환하여 출력한다.
  - **atoA** 함수를 N회 반복해서 호출함
- 힌트: `c = atoA( ____ );`

입력 예시

출력 예시

10 <---- N  
He1l0WorLd

HELLOWORLD

[ 문제 2 ] 양의 정수 N과 M을 입력받아, 2 ~ N 사이의 소수를 구하여 각 줄에 M개씩 출력하는 프로그램을 작성하라.

- **is\_prime** 함수를 사용하여 소수를 구한다.

인자는 한 개의 정수형 변수

반환은 0 또는 1을 반환 함 (0은 소수가 아님, 1은 소수 임)

- **print\_prime** 함수

인자는 두 개의 정수형 변수 N과 M

**is\_prime** 함수에 2부터 N의 값을 전달, 반환 결과 소수이면 출력 (각 줄은 M개의 소수)

- **main** 함수

N과 M을 입력받는다.

`print_prime` 함수 호출

입력 예시

출력 예시

50 5 <---- N, M

2 3 5 7 11  
13 17 19 23 29  
31 37 41 43 47

[ 문제 3 ] 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	↳ 영문자 이외의 문자는 제외
	1	↳ Cat 1개

입력 예시 2

출력 예시 2

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

[ 문제 4 ] 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 작성한다.

- **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 5 ]** 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

23303843

출력 예시 1

23084

↳ 뒤에 나온 3제거

입력 예시 2

891571219

출력 예시 2

891572

↳ 뒤에 나온 1, 9 제거

**[ 문제 6 ] 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 7 ]** 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 자리수의 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
---------	------------------	---