

〈고급 C프로그래밍 및 실습〉 13장 고급 문법 실습 문제

※ 문제에 대한 안내

- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정하라.
- 특별한 언급이 없으면, 각 줄의 맨 앞과 맨 뒤에는 공백을 출력하지 않는다.
- 출력 예시에서 □는 각 줄의 맨 앞과 맨 뒤에 출력되는 공백을 의미한다.
- 입출력 예시에서 ↦ 이 후는 각 입력과 출력에 대한 설명이다.

2절 [문제 1] 두 개의 양의 정수를 입력받고, 이 두 정수의 최대공약수를 출력하는 프로그램을 작성하시오. 반드시 **재귀함수**를 사용하시오.

입력 예시 1

출력 예시 1

366 60	6
--------	---

Hint: 유클리드 호제법으로 최대공약수 계산하기

(예) 입력 (12, 8) 1단계 (8, 12-8=4) 2단계 (4, 8-4=4) 3단계 (4, 4) 최대공약수는 4
(12 - 8 = 큰 수에서 작은 수 뺄셈)

2절 [문제 2] 오일러의 수 e는 자연 로그의 밑수로 사용된다. 이 값은 다음과 같은 식에 의하여 근사치를 구할 수 있다. 오일러 수를 계산하는 프로그램을 작성하시오.

- 소수점 이하 **여섯째 자리**까지 화면에 출력 하시오.

$$= 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{n!}$$

입력 예시 1

출력 예시 1

30 ↦ 계산할 수	2.718282
------------------	----------

Hint: factorial() 함수 호출하여 사용

factorial() 함수는 **재귀함수**로 구현

2절 [문제 3] **재귀함수**를 이용하여 사용자가 입력한 N 번째 피보나치 수열의 값을 구하는 프로그램을 작성하시오. 여기서, 피보나치 수열은 다음과 같은 점화식으로 표현된다.

$$F_n = \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F_{n-1} + F_{n-2} & \text{if } n > 1. \end{cases}$$

즉, 피보나치 수 F_n 은 0과 1로 시작되며, 다음 피보나치 수는 바로 앞의 두 피보나치 수의 합이

된다. (즉 $n=0$ 와 $n=1$ 일 때는 예외적이고 $n=2$ 부터 정상적으로 앞 두 수의 합이다)

$n = 0, 1, 2, 3, 4, 5, 6, 7, \dots$ 에 해당하는 피보나치 수는 $F_n = 0, 1, 1, 2, 3, 5, 8, 13, \dots$ 이 된다.

n	0	1	2	3	4	5	6	7	...
피보나치수	0	1	1	2	3	5	8	13	...

입력 예시 1

출력 예시 1

6 \mapsto n	8
---------------	---

1절 [문제 4] 0~255 범위(8 bits 표현 범위)의 정수 4개를 입력 받아, 1개의 unsigned int 형으로 변환하여 출력하는 프로그램을 작성하시오. 입력 순서대로 정수 하나씩(8 bits) 순차적으로 결합하여 unsigned int형의 정수 한 개를 만든다.

- (예시 2)에서 4 bytes 크기의 정수 한 개를 화면에 출력한다. 출력할 4 bytes 중에서, 15는 첫 번째 바이트를 차지하고, 4는 두 번째 바이트, 20은 세 번째 바이트, 3은 네 번째 바이트를 차지한다.

입력 예시 1

출력 예시 1

1 1 1 1 \mapsto 정수 4개	16843009 \mapsto 16진수: 0x01010101
-------------------------	-------------------------------------

입력 예시 2

출력 예시 2

15 4 20 3 \mapsto 정수 4개	251925507 \mapsto 16진수: 0xF041403
---------------------------	-------------------------------------

4절(심층) [문제 5] int형 변수 두 개 x, y 와 두 변수를 가리키는 포인터 px, py 를 선언하고, 포인터 변수에 저장된 주소를 바꾸는 함수 `pswap()`을 만들어서 `main()` 함수 내에서 포인터가 가리키는 값을 출력하시오.

1) `pswap` 함수

- 인자: 포인터를 가리키는 포인터 두 개 (즉 이중포인터), `ppx, ppy`
- 포인터에 담긴 수(주소)를 서로 교환
- 반환값: 없음

2) `main` 함수

- int형 변수 두 개 x, y 와 두 변수를 가리키는 포인터 px, py 를 선언
- 그래서 px 에는 x 의 주소가 저장되어 있고, py 에는 y 의 주소가 저장된다.
- px 의 주소와 py 의 주소를 실인자로 하는, 함수 `pswap()`을 호출하여, px 에 저장된 주소와 py 에 저장된 주소를 교환한다.
- 그래서 px 에는 y 의 주소가 저장되게 하고, py 에는 x 의 주소가 저장된다.
- 포인터 px, py 가 가리키는 값을 출력하시오. 키보드로부터 입력된 값의 순서가 바뀌어 출력된다.

입력 예시 1

출력 예시 1

1 5 ↦ x y	5 1
----------------	-----

5절(심층) [문제 6] 2차원 배열의 각 원소 값을 해당 원소의 두 인덱스 값의 합으로 하는 프로그램을 다음과 같이 작성 하시오.

- ABC() 함수를 다음과 같이 작성한다.
 - 함수 ABC()는 "배열의 이름"을 인자로 받고, 두 정수 k1, k2를 인자로 전달 받는다. 인자가 3개 이다. k1은 인덱스의 개수, k2도 인덱스의 개수 이다. (강의노트 p. 44, 57)
 - `void ABC(int (*p)[20], int k1, int k2);`
 - 각 원소의 값을 해당 원소의 두 인덱스 값의 합으로 수정한다.
 - 배열 표기 대신 포인터 표기를 사용한다.
 - Hint: 두 인덱스의 합을 2차원 배열에 저장은 다음과 같다.
 - `*(p+i)[j]=i+j;`
- main() 함수를 다음과 같이 작성한다.
 - N X N의 2차원 배열 arr을 선언하고, 배열의 원소값을 0으로 초기화 한다. N은 20 이하
 - 함수 ABC()를 호출한다.
 - 배열 arr의 내용을 화면에 출력한다.

입력 예시 1

출력 예시 1

3 ↦ N=3	<input type="checkbox"/> 0 <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4
--------------	--

6절(심층) [문제 7] 1단계를 먼저 구현하고, 2단계를 구현하시오.

- 1) 정수를 키보드로부터 입력 받아 int형 변수 A에 저장한다. int형 변수 A의 주소를 void 포인터에 저장한다. void 포인터를 사용하여 int형 변수에 담긴 값을 화면에 출력한다.
 - 즉 void 포인터를 int형 포인터로 형변환 한 후, 역참조연산자 * 를 사용하여 포인터가 가리키는 변수에 저장된 값을 화면에 출력한다. (강의노트 p. 64 참조하시오)
- 2) 두 정수 x, y의 합을 출력하기 (강의노트 p. 68 참조하시오)
 - 합을 구하는 함수 add 를 정의하시오.
 - main 함수에서
 - 함수 포인터 fp를 선언하고 함수 add와 연결하시오.
 - 두 정수를 키보드로부터 입력 받으시오.
 - 함수 포인터 fp를 사용하여 두 변수의 합을 화면에 출력하시오.

입력 예시 1

6	↦ A
12 15	↦ x y

출력 예시 1

6	↦ void 포인터가 가리키는 변수 A의 값
27	↦ 12+15=27

6절(심층) [문제 8] 배열의 크기 3인 함수 포인터 배열을 만들어 각각 더하기와 빼기, 그리고 곱하기를 수행하는 함수를 각각 저장한다. 더하기 빼기 곱하기를 수행하는 프로그램을 작성하시오. (강의 노트에 있는 (예제5)와 그 다음 페이지 참고하시오)

1) 더하기, 빼기, 곱하기 함수를 정의하시오. 함수원형은 다음과 같다.

```
int add(int, int);
int sub(int, int);
int multiply(int, int);
```

2) main 함수

- 배열의 크기 3인 함수 포인터 배열 handle 을 선언하고, 함수 포인터 배열의 각 원소와 1)에서 정의한 각 함수와 연결하시오. (여기서는 함수 포인터 배열을 함수 이름으로 초기화 한다)

```
int (*handle[3])(int, int) = { add, sub, multiply };
```

- 정수 두 개를 키보드로부터 읽어 들이기
- 함수포인터를 이용하여 세 개의 함수를 호출한다.
- 세 가지 연산 결과를 화면에 출력한다.

입력 예시 1

2 4	↦ x y
-----	-------

출력 예시 1

6 -2 8	↦ 2+4 2-4 2×8
--------	---------------

입력 예시 2

10 7

출력 예시 2

17 3 70
