



과제 8-1. 사칙 연산

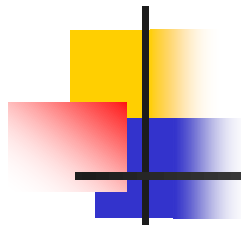
8-1 :

- 다음 페이지 프로그램 은 포인터를 이용하여 덧셈을 하는 프로그램이다.
- 이 프로그램을 수정하여 뺄셈, 곱셈, 나눗셈, 그리고 나머지 연산을 실행하는 프로그램을 작성하라.

■ 실행 결과

```
Enter the first number : 15
Enter the second number : 51

15 + 51 is 66
15 - 51 is -36
15 * 51 is 765
15 / 51 is 0
15 % 51 is 15
계속하려면 아무 키나 누르십시오 . . .
```



Add Two Numbers Using Pointers

```
1  /* This program adds two numbers using pointers to
2     demonstrate the concept of pointers.
3     Written by:
4     Date:
5  */
6  #include <stdio.h>
7
8  int main (void)
9  {
10 // Local Declarations
11     int  a;
12     int  b;
13     int  r;
14     int* pa = &a;
15     int* pb = &b;
16     int* pr = &r;
17
18 // Statements
19     printf("Enter the first number : ");
20     scanf ("%d", pa);
21     printf("Enter the second number: ");
22     scanf ("%d", pb);
23     *pr = *pa + *pb;
24     printf("\n%d + %d is %d", *pa, *pb, *pr);
25     return 0;
26 } // main
```



과제 8-2. 최소공배수, 최대공약수 계산

8-2 :

두 개의 정수를 파라미터로 받아, 두 수의 최대공약수와 최소공배수를 계산하는 함수를 구현.

■ 함수 선언 형태 : *void calGcdLcm (int num1, int num2, int * gcd, int * lcm);*

- 최대공약수 계산은 다음 페이지 이미지에서 설명한 것 처럼, modulo ‘%’를 이용하여 한 수를 다른 수로 나눈 나머지를 이용하는 유클리디안 호제법을 사용한다.
- 나머지가 0이 되면 최대 공약수를 찾은 것임.
- 최소공배수는 $lcm = (num1 * num2) / gcd$ 으로 계산.

■ 메인 프로그램에서는 이 함수를 이용하여 **standard input**으로 두 개의 정수를 입력 받고, **standard output**으로 최대 공약수와 최소 공배수를 출력하는 프로그램을 작성한다.

■ 제약조건

- **calGcdLcm** 함수 안에서 결과를 **print**하지 마시오.
- 전역변수를 사용하지 마시오. (함수 안에서 지역변수만 사용 가능)

과제 8-2. 최소공배수 최대공약수 계산

- 아래와 같은 결과가 나오도록 한다.

Example

```
Enter the 1st number : 247
Enter the 2nd number : 39
```

```
GCD of 247 and 39 is 13
LCM of 247 and 39 is 741
```

```
Enter the 1st number : 18
Enter the 2nd number : 45
```

```
GCD of 18 and 45 is 9
LCM of 18 and 45 is 90
```

```
Enter the 1st number : 45
Enter the 2nd number : 18
```

```
GCD of 45 and 18 is 9
LCM of 45 and 18 is 90
```

<예>

gcd(710,68)을 구해보자.

$$710 = 68 \cdot 10 + 30$$

$$68 = 30 \cdot 2 + 8$$

$$30 = 8 \cdot 3 + 6$$

$$8 = 6 \cdot 1 + 2$$

$$6 = 2 \cdot 3 + 0$$

즉, 최종값은 2이다.



과제 8-3. 시간변환 함수의 이용

과제 8-3: 강의자료의 prog-8 secToHours 함수를 이용하여

시간 array에 임의의 숫자를 입력 받는 함수를 만든다.

- 함수 선언 형태 : *int getTime (long time[], int max_size);*
 - Time[] 배열에 입력된 시간의 개수를 반환한다.
 - 최대 입력 받는 시간의 개수는 max_size로 한다.

초 단위 시간 값을 시, 분, 초로 변환하는 함수는 앞의 프로그램 9-8을 이용함

- *void secToHours (long time, int *hours, int *minutes, int *seconds);*

메인 프로그램 제약 사항:

- *getTime* 함수를 이용하여 주어진 시간 배열에 여러 개의 정수를 입력 받는다. 단 0을 입력하면 입력을 종료한다.
- 배열의 최대 크기는 10으로 잡는다.
- 입력된 배열에 저장된 각각의 시간 값을 *secToHours* 함수를 이용하여 시간/분/초로 변환하여 차례로 출력한다.



과제 8-3. 시간변환 함수의 이용

■ 실행 결과

```
변환할 시간을 입력하기 바랍니다.6405
변환할 시간을 입력하기 바랍니다.3463
변환할 시간을 입력하기 바랍니다.3601
변환할 시간을 입력하기 바랍니다.3504
변환할 시간을 입력하기 바랍니다.5332
변환할 시간을 입력하기 바랍니다.0
6405 = 1:46:45
3463 = 0:57:43
3601 = 1:00:01
3504 = 0:58:24
5332 = 1:28:52

END OF PROGRAM
계속하려면 아무 키나 누르십시오 . . .
```