

## LAB9

### ■ LAB9\_1(매개변수를 가지는 void 함수의 연습)

아래의 실행결과를 갖는 프로그램을 printManyStars(...) 함수를 사용하여 작성하시오.

<pre>#include &lt;stdio.h&gt; void printManyStars(int num); // 함수의 원형 int main(void) {     printManyStars(3);     // 함수의 호출부분 추가  }  void printManyStars(int num) // 함수의 정의, num만큼 *를 출력한다 {     int i;      // 여기에 num만큼 *를 출력하는 코드를 작성하시오. }</pre>	실행예)  *** **** *****
--	----------------------------------

### ■ LAB9\_2(매개변수를 가지는 void 함수의 연습)

양의 정수를 입력받아서 그것의 Factorial 을 구하는 프로그램을 작성하시오.

<pre>#include &lt;stdio.h&gt; void get_factorial(int n); // 함수의 원형 int main(void) {     int num;      printf("Enter the number: ");     scanf("%d", &amp;num);      get_factorial(num);  }  void get_factorial(int n) // 함수의 정의 {     //n에 대한 factorial 값을 구하여 출력한다. }</pre>	실행예:(밑줄은 입력)  Enter the numbers: 5 5! = 120
--	--

### ■ LAB9\_3 배수 구하기(매개변수를 가지는 void 함수의 연습)

정수와 배수의 개수를 매개변수로 전달받아 정수의 배수를 출력하는 함수를 작성하여 프로그램을 완성하시오.

- 정수로 0 보다 작거나 같은 값이 입력되면 프로그램을 종료한다.
- 정수의 배수는 10 개만 출력되도록 한다.

<pre>#include &lt;stdio.h&gt; void multiples(int num, int count) ;  int main(void) {     int num;      printf("Enter the number: ");     scanf("%d", &amp;num);     while (num &gt; 0) {         multiples(num, 10);         printf("Enter the number: ");         scanf("%d", &amp;num);     }     return 0 ; }  void multiples(int num, int count) {     //num에 대한 배수를 구해 출력한다. }</pre>	실행결과예 : (밑줄은 입력)  Enter the number: 5 5 10 15 20 25 30 35 40 45 50 Enter the number: 11 11 22 33 44 55 66 77 88 99 110 Enter the number: 0
---	--

### ■ LAB9\_4 약수와 약수의 갯수 구하기(매개변수를 가지는 void 함수의 연습)

정수를 입력받아 그 정수의 약수와 약수의 개수를 구해서 출력하는 함수를 이용하여 프로그램을 작성하시오.

- 정수로 0이 입력되면 프로그램을 종료한다.

<pre>#include &lt;stdio.h&gt; void divisors(int n); //함수 원형  int main(void) {     int num;      printf("양의 정수?");     scanf("%d", &amp;num);      while (num != 0) {         divisors(num);         printf("양의 정수?");         scanf("%d", &amp;num);     }     return 0; }  void divisors(int n) //함수 정의 {      //n에 대한 약수와 약수의 개수를 구해 출력한다. }</pre>	실행결과예 : (밑줄은 입력)  양의 정수?5 5의 약수: 1 5 => 총 2개 양의 정수?36 36의 약수: 1 2 3 4 6 9 12 18 36 => 총 9개 양의 정수?0
--	--

### ■ LAB9\_5 윤년 구하기(매개변수를 가지는 value returning 함수의 연습)

연도를 매개변수로 전달받아 윤년인지 판단하는 함수를 작성하고 이를 이용하여 2000 년부터 2100 년 사이에 있는 윤년을 모두 구하여 출력하는 프로그램을 작성하시오.

- 결과는 한 줄에 10 개씩 출력한다.

#### □ 윤년구하는 판단 기준

- 4 로 나누어지고 100 으로 나누어지지 않는 해
- 400 으로 나누어지는 해

<pre>#include &lt;stdio.h&gt; int is_leap_year(int year) ; int main(void) {     변수선언      for (i = 2000; i &lt;= 2100; i++)         // 여기에 필요한 코드를 완성하시오. }  int is_leap_year(int year) {     //윤년인지 아닌지 판단하여 윤년이면 1     그렇지 않으면 0 을 return 한다. }</pre>	실행결과예 :  2000 2004 2008 2012 2016 2020 2024 2028 2032 2036 2040 2044 2048 2052 2056 2060 2064 2068 2072 2076 2080 2084 2088 2092 2096
---	---

- **LAB9\_6** 소수구하기(매개변수를 가지는 value returning 함수의 연습)  
num 이 소수이면 1 을 아니면 0 을 반환하는 함수 isPrime()을 작성하여 프로그램을 완성하시오.

<pre>#include &lt;stdio.h&gt; int isPrime(int n); int main(void) {     //필요한 코드 완성하시오  }  int isPrime(int n) {     // 소수이면 1 을 아니면 0 을 return 하는 코드 작성  }</pre>	실행결과 예1 : (밑줄은 입력)  Enter a number : <u>5</u> 소수입니다.
---	---

HW 9

- **HW9\_1**(매개변수를 가지는 void 함수의 연습)  
print5Chars 함수를 사용하여 아래처럼 특정문자를 n개만큼 출력하는 프로그램을 작성하시오.

<pre>// 매개변수를 가지는 void 함수를 사용한 프로그램 연습 #include &lt;stdio.h&gt; void print5Chars(char ch);    // 함수의 원형 int main(void) {     print5Chars('*', 5);      // 함수의 호출  }  // 함수의 정의: 주어진 문자를 n번 출력하는 함수를 정의 void print5Chars(char ch, int n) {     // 여기에 ch를 n개 출력하는 코드를 작성하시오.  }</pre>	실행예)  ***** @@@@@@@@ 1111111111
--	---

- **HW9\_2**(리턴값과 매개변수를 가지는 함수 연습)  
아래의 실행결과를 갖는 프로그램을 sumMToN 함수를 사용하여 작성하시오.

<pre>#include &lt;stdio.h&gt; int sumMToN(int m, int n); // 함수의 원형  int main(void) {     printf("%d부터 %d까지의 합은 %d\n", 2, 5, __ );     printf( );     printf( ); }  int sumMToN(int m, int n) // 함수의 정의 {      // 이곳에 m부터 n까지의 합을 계산하여     // 그 합을 return하는 함수 정의부분을 코딩한다  }</pre>	실행 결과예)  2부터 5까지의 합은 14 3부터 7까지의 합은 25 2부터 10까지의 합은 54
---	--

- **HW9\_3**(리턴값과 매개변수를 가지는 함수 연습)  
정수를 3 개를 입력받아서 그 중에서 가장 큰 수를 찾아 출력하는 프로그램을 작성하시오.


-가장 큰 수를 찾는 작업은 별도의 함수를 만들어서 사용한다.

<pre>int getMax(int x, int y, int z);    // 함수 원형 int main(void) {     int num1, num2, num3;     int bigNumber;      printf("Enter a numbers: ");     scanf("%d %d %d", &amp;num1, &amp;num2, &amp;num3);      bigNumber = getMax(num1, num2, num3);      printf("Big number is %d.\n", bigNumber);     return 0; }  int getMax(int x, int y, int z) {     //x, y, z에서 가장 큰 수를 찾아서 리턴한다.  }</pre>	실행결과 예: (밑줄은 입력)  Enter three numbers: 5 10 7 Big number is 10.
---	--

■ HW9\_4

num이 소수이면 1을 아니면 0을 반환하는 함수 isPrime()을 이용하여 2부터 시작해서 차례대로 10개의 소수를 구하여 출력하는 프로그램을 완성하시오.

☞ 힌트 : 감시값 제어 반복문 사용

<pre>#include &lt;stdio.h&gt; int isPrime(int n); //함수 원형 int main(void) {     //여기에 적절한 코드를 작성하시오. }  int isPrime(int n) {     //여기에 n이 소수이면 1을 그렇지 않으면     0을 return하는 코드를 작성하시오. }</pre>	
---	---

■ HW9\_5(누산기 : 연산의 결과에 연속적으로 계산을 수행하는 계산기)

- 입력받은 수식의 계산 결과를 출력하는 프로그램을 작성하시오.
- 연산자와 피연산자 하나를 매개변수로 전달받아 이전 연산의 결과와 계산을 수행하고 그 결과를 출력하는 함수를 정의하여 사용한다.
  - "0 0 "이 입력되면 프로그램을 종료한다.
  - 지역변수만 이용하여 작성한다.(전역변수를 사용하지 않는다)

<pre>#include &lt;stdio.h&gt; int accumulator(char op, int n, int prevResult);  int main(void) {     int result = 0;     printf("%d ", result);      //적절한 코드 입력 }  int accumulator(char op, int n, int prevResult) {     // 이전 연산의 결과와 계산을 수행하는 코드 작성 }</pre>	
--	---