

# 7주차 과제

프로그래밍 연습

# Problem 1

특정 문자로 이루어진 다이아몬드를 만드는 함수 작성

사용자로 부터 임의의 문자와 숫자를 입력 받은 후 입력받은 값을 사용하여 다이아몬드를 출력

- 입력받는 값은 홀수 정수 (다이아몬드 모양의 중간 길이 사이즈)
- 사용자에게 문자를 입력받고 출력하는 기능을 각각의 함수로 구현하여 사용

## Output

표현문자를 입력하세요

>>\*

크기를 입력하세요

>>9

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*

\*

# Problem 2

## 소수 찾기 실습 코드 업그레이드

### is\_prime() 함수를 수정하여 실행속도를 빠르게하기

- 실습자료 p.35~36에 있는 소수찾기 코드를 사용
- 해당코드는 검사하는 숫자가 매우 큰 경우 비효율적 ex) 1,000,000 -> 백만번 반복
- TIP) 1보다 크고 n보다 작은 숫자 중 약수가 하나라도 발견되는 경우 이미 n은 소수가 아니라고 생각
- time.h를 include하여 코드 실행시간 측정
- 기존 is\_prime()함수를 사용하였을때의 시간, 수정한 is\_prime\_upgrade()함수를 사용하였을때의 시간을 각각 출력하여 실행속도가 빨라짐을 확인

### Output

정수를 입력하십시오

>>23

23은 소수입니다

기존 함수 측정시간:~s

수정 함수 측정시간:~s

# Problem 2 Sample code

```
#include <stdio.h>
#include <time.h>

int is_prime(int);
int get_integer();

int main(void)
{
    clock_t start, end;
    float res;
    int n, result;

    n=get_integer();
    start = clock();
    result = is_prime(n);
    end = clock();

    res = (float)(end-start)/CLOCKS_PER_SEC;

    if(result == 1)
        printf("%d은 소수입니다\n",n);
    else
        printf("%d은 소수가 아닙니다\n",n);

    printf("소요시간: %.3f \n",res);

    return 0;
}
```

정수를 입력하시오 : 100000000  
100000000은 소수가 아닙니다  
소요시간 : 0.726

# Problem 3

사용자로부터 입력을 받아  
특정 기능을 수행하는 프로그램 작성

1번 - 문자를 하나 입력받아서 대소문자 여부를 출력

2번 - 숫자를 하나 입력받아서 홀수 짝수 확인

3번 - N factorial 구하기 (N!)

0번 - 프로그램 종료

- 사용자가 0을 입력하여 프로그램 종료를 하기 전까지 특정 기능을 수행해야함
- 프로그램 종료를 제외한 모든 기능은 함수형태로 구현
- 정해진 입력을 제외한 다른 입력을 받았을 경우에 대응할 수 있도록 구현

## Output

1. 대소문자 확인
2. 홀수 짝수 확인
3. N 팩토리얼 구하기(N!)
0. 종료

>>1

영문자를 입력하세요: A

'A'는 대문자 입니다.

1. 대소문자 확인
2. 홀수 짝수 확인
3. N 팩토리얼 구하기(N!)
0. 종료

>>2

정수를 입력하세요:100

100는(은) 짝수입니다.

1. 대소문자 확인
2. 홀수 짝수 확인
3. N 팩토리얼 구하기(N!)
0. 종료

>>3

정수를 입력하세요:7

7!은 5040입니다

1. 대소문자 확인
2. 홀수 짝수 확인
3. N 팩토리얼 구하기(N!)
0. 종료

>>9

올바른 값을 입력하세요

1. 대소문자 확인
2. 홀수 짝수 확인
3. N 팩토리얼 구하기(N!)
0. 종료

>>0