

```
2019112602@linuxserver1:~/HW$ vi coding_test4.c
2019112602@linuxserver1:~/HW$ vi coding_test5.c
2019112602@linuxserver1:~/HW$ gcc coding_test5.c
2019112602@linuxserver1:~/HW$ ./a.out
연도를 입력 하시오 : 2012
윤년 여부 = 1
2019112602@linuxserver1:~/HW$
```

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

```
int main(void)
{
```

```
    int year, result;
```

```
    printf("연도를 입력 하시오 : ");
    scanf("%d", &year);
```

```
    result = ((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0);
```

```
    printf("윤년 여부 = %d \n", result);
```

```
    return 0;
```

```
}
```

```
4.cs.dongguk.edu x +
#include <stdio.h>

int main(void)
{
    int room_width = 320;
    int room_height = 200;
    int tile_size = 30;

    int tiles_needed = (room_width / tile_size + (room_width % tile_size != 0)) * (room_height / tile_size + (room_height % tile_size != 0));
    printf("필요한 타일 개수: %d개\n", tiles_needed);

    return 0;
}
```

```
2019112602@linuxserver1:~/HW$ vi coding_test3.c
2019112602@linuxserver1:~/HW$ vi coding_test4.c
2019112602@linuxserver1:~/HW$ gcc coding_test4.c
2019112602@linuxserver1:~/HW$ ./a.out
필요한 타일 개수 : 77개
2019112602@linuxserver1:~/HW$ █
```



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

```
int main(void)  
{
```

```
    int money, price, change;
```

```
    printf("투 입 한 돈 : ");  
    scanf("%d", &money);
```

```
    printf("를 건 값 : ");  
    scanf("%d", &price);
```

```
    change = money - price;  
    printf("거 스 름 돈 : %d\n", change);
```

```
    int coin100s = change / 100;  
    change = change % 100;
```

```
    int coin10s = change / 10;  
    change = change % 10;
```

```
    printf("100원 통 전 의 개 수 : %d\n", coin100s);
```

```
    printf("10원 통 전 의 개 수 : %d\n", coin10s);
```

```
    printf("1원 통 전 의 개 수 : %d\n", change);
```

```
    return 0;
```

```
}
```

```
2019112602@linuxserver1:~/HW$ ./a.out
투입한 돈 : 1000
เหรียญ값 : 271
거스름돈 : 729
100원 동전의 개수 : 7
10원 동전의 개수 : 2
1원 동전의 개수 : 9
2019112602@linuxserver1:~/HW$
```



4. cs.dongguk.edu



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

```
int main(void)
```



```
    int num, value, digit_sum = 0;
```

```
    printf("3자 리 숫 자 를 입 력 하 시 오 : ");
```

```
    scanf("%d", &num);
```

```
    value = num;
```

```
    // 각 자 리 수 를 분 리 하 여 합 산
```

```
    digit_sum += num % 10; // 1의 자 리
```

```
    num /= 10;
```

```
    digit_sum += num % 10; // 10의 자 리
```

```
    num /= 10;
```

```
    digit_sum += num % 10; // 100의 자 리
```

```
    printf("%d의 자 릇 수 합 은 %d입 니 다 .\n", value, digit_sum);
```

```
    return 0;
```



```
2019112602@linuxserver1:~/HW$ gcc coding_test2.c
2019112602@linuxserver1:~/HW$ ./a.out
3자 리 숫 자 를 입 력 하 시 으 : 365
365의 자 린 수 합 은 14입 니 다 .
2019112602@linuxserver1:~/HW$ █
```



```
2019112602@linuxserver1:~/HW$ ./a.out
```

```
6383는 days
```

```
17는 years
```

```
5는 months
```

```
29는 remaining_days
```

```
하루 출퇴근 거리 : 60 km
```

```
달 까 지 의 거 리 : 383000 km
```

```
17년 5월 29일 후 출퇴근 거 리 가 달 까 지 의 거 리 가 됩 니 다 .
```

```
2019112602@linuxserver1:~/HW$ █
```



```
printf("달까지의 거리 : %d\n", distance);
```

```
printf("%d년 %d월 %d일 후 출퇴근 거리가 달까지의 거리가 됩니다.\n", y, m, final_d);
```

```
return 0;
```

```
}
```

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

```
int main(void)
```

```
{
```

```
const int daily_commute = 60; // 하루 출퇴근 거리 (km)
```

```
const int distance_to_moon = 383000; // 달까지의 거리 (km)
```

```
int days = ((double)distance_to_moon / daily_commute + 1);
```

```
int years = ((double) days / 365);
```

```
int months = ((double) (days % 365) / 30);
```

```
int remaining_days = ((double)(days % 365) % 30);
```

```
printf("하루 출퇴근 거리 : %d km\n", daily_commute);
```

```
printf("달까지의 거리 : %d km\n", distance_to_moon);
```

```
printf("%d년 %d월 %d일 후 출퇴근 거리가 달까지의 거리가 됩니다.\n", years, months, remaining_days);
```

```
return 0;
```

```
}
```

```
*/
```

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

```
int main(void)
```

```
{
```

```
const int daily_commute = 60; // 하루 출퇴근 거리 (km)
```

```
const int distance_to_moon = 383000; // 달까지의 거리 (km)
```

```
int days = (int)((double)distance_to_moon / daily_commute );
```

```
int years = days / 365;
```

```
int months = (days % 365) / 30;
```

```
int remaining_days = (days % 365) % 30 + 1;
```

```
printf("%d= days\n", days);
```

```
printf("%d= years\n", years);
```

```
printf("%d= months\n", months);
```

```
printf("%d= remaining_days\n", remaining_days);
```

```
printf("하루 출퇴근 거리 : %d km\n", daily_commute);
```

```
printf("달까지의 거리 : %d km\n", distance_to_moon);
```

```
printf("%d년 %d월 %d일 후 출퇴근 거리가 달까지의 거리가 됩니다.\n", years, months, remaining_days);
```

```
return 0;
```

```
}
```

```
#include <stdio.h>

int main (void)
{
    const int salary = 60; // 본영 왕복 출퇴근 거리가 60km라고 했지 편도가 60km이라고
    //문득에서 주어진지 않았습니다. 하루에 60km를 이동한다고 생각하여 했습니다
    const int distance = 383000;

    int d = distance/salary + 1;
    int y = d/365;
    int m = (d%y)/30;
    int final_d = (d%y)%30;

    printf("왕복 출퇴근 거리 : %d \n", salary);
    printf("달까지의 거리 : %d \n", distance);

    printf("%d년 %d월 %d일 후 출퇴근 거리가 달까지의 거리가 됩니다.\n", y, m, final_d);

    return 0;
}
```

```
#include <stdio.h>

int main (void)
{
    const int daily_commute = 60; // 하루 출퇴근 거리 (km)
    const int distance_to_moon = 383000; // 달까지의 거리 (km)
    int days = ((double)distance_to_moon / daily_commute + 1);
    int years =((double) days / 365);
    int months = ((double) (days % 365) / 30);
    int remaining_days = ((double)(days % 365) % 30);

    printf("하루 출퇴근 거리 : %d km\n", daily_commute);
    printf("달까지의 거리 : %d km\n", distance_to_moon);
    printf("%d년 %d월 %d일 후 출퇴근 거리가 달까지의 거리가 됩니다.\n", years, months, remaining_days);

    return 0;
}
*/
```

```
#include <stdio.h>

int main(void)
{
    const int daily_commute = 60; // 하루 출퇴근 거리 (km)
    const int distance_to_moon = 383000; // 달까지의 거리 (km)
    int days = (int)((double)distance_to_moon / daily_commute );
```

문제 1. (4점) 다음 코드를 분석하여, 왜 b의 값이 예상과 다를 수 있는지 설명하시오. 또한, 변수 b에 있는 메모리 초기화의 중요성을 설명하시오.

```
#include <stdio.h>

int main() {
    int a = 5;
    int b;

    b = a + b;

    printf("b의 값: %d\n", b);

    return 0;
}
```

문제 1 답 : 변수를 선언할 때 0이나 기타 상수로 초기화를 하지 않으면 이전의 쓰레기 값이 남아서 계산이 제대로 되지 않을 수 있습니다. 특히 지금 상황에서 b값을 0으로 지정하지 않았기에, 5값을 가진 a와 더한다 하더라도 5가 나온다는 보장이 없습니다.

문제 2. (5 점) 다음 코드에서 const 상수의 역할을 설명하고, 이 코드를 통해 상수를 사용하는 이유와 상수의 변경 가능성에 대해 논의하시오. 상수를 제대로 사용하는 방법을 설명하시오.

```
#include <stdio.h>

int main() {
    const int max_value = 100;
    int a = 50;

    a = a + max_value;
    max_value = 200; // 이 줄에서 어떤 일이 발생하나요?

    printf("a 의 값: %d\n", a);
    printf("max_value 의 값: %d\n", max_value);

    return 0;
}
```

문제 2 답 : const함수는 해당 변수에 상수를 입력하는 기능을 합니다. 그런데 8번째 줄에서 임의로 max_value값을 200으로 다른 수를 집어넣었기에 오류가 발생합니다. 한번 상수로 선언된 변수값은 바꿀 수 없고 고정을 시킵니다. 상수를 변경하려면 처음부터 const 함수를 빼 버리거나, 아니면 4번째 처음의 줄에서부터 100 대신 200을 집어넣어야 합니다

문제 4. (8 점) 다음 코드를 보고 연산자 우선순위에 의해 연산이 어떤 순서로 진행되는지 설명하고, 왜 결과가 일부 예상과 다르게 나올 수 있는지 분석하시오.

```
#include <stdio.h>

int main() {
    int a = 5;
    int b = 10;
    int c;

    c = a + b * 2 / 5 - 1;

    printf("c의 값: %d\n", c);

    return 0;
}
```

우선 c의 값을 0이나 다른 수로 초기화하지 않았으므로 쓰레기 값이 섞여있을 위험이 있습니다. 컴퓨터에서의 연산 또한 마찬가지로 수학의 연산 순서와 동일하기에

우선 $b * 2 / 5$ 순으로 연산이 된 이후, 맨 앞에서부터 $a + ? - 1$ 순으로 연산이 됩니다

괄호를 쓰면 일일이 이를 신경쓸 필요 없기 때문에 가독성과 편리를 위해 괄호를 쓰는 것이 좋습니다