

Computer Programming Assignment #2 답지

● Problem 1 (10점)

1) 최대공약수(GCD)를 구하는 프로그램을 작성한다

- 두 개의 정수를 입력 받아 Euler 알고리즘을 이용하여 최대공약수가 계산되는 과정을 보이고 마지막으로 입력된 두 정수의 최대공약수를 출력한다. 이때, 반복문을 이용하여 구현한다.

- GCD를 찾는 방법

I. 두 개의 정수 a, b 가 있을 때, $a > b > 0$ 이면 $a = k_1 * b + r_1$ 로 나타낼 수 있다. 이때, $\text{GCD}(a, b) = \text{GCD}(b, r_1)$ 이라는 점을 이용해서 다음과 같이 a, b 의 최대공약수를 계산할 수 있다.

$$b = k_2 * r_1 + r_2$$

$$r_1 = k_3 * r_2 + r_3$$

...

$$r_n = k_{n+1} * r_n + r_{n+1}$$

II. r_{n+1} 이면 $\text{GCD}(r_{n-1}, r_n) = r_n$ 이므로 $\text{GCD}(a, b) = \text{GCD}(b, r_1) = \text{GCD}(r_1, r_2) = \dots = \text{GCD}(r_{n-1}, r_n)$ 으로부터 $\text{GCD}(a, b) = r_n$ 을 얻을 수 있다. (실행 예 참고)

두 정수를 입력하시오.: 273 98

$$273 = 2 * 98 + 77$$

$$98 = 1 * 77 + 21$$

$$77 = 3 * 21 + 14$$

$$21 = 1 * 14 + 7$$

$$14 = 2 * 7 + 0$$

273과 98의 최대공약수는 7이다.

```

#include<stdio.h>

int gcd(int a, int b) {

    int small, large, n;

    // a와 b중 더 큰 수가 large에, 작은 수가 small에 할당됩니다.
    large = a > b ? a : b;
    small = a < b ? a : b;

    // small이 0이 되는 순간의 large가 최대 공약수이므로, 리턴합니다.
    while (small != 0) {
        printf("%d = %d*%d + %d\n", large, large / small, small, large % small);
        n = large % small;
        large = small;
        small = n;
    }
    return large;
}

int main() {

    int a, b;
    int result;

    printf("두 정수를 입력하시오.: ");
    scanf_s(" %d %d", &a, &b);

    // gcd 함수에서 if (small == 0)에서 return 된 값이 저장됩니다.
    result = gcd(a, b);

    printf("%d과 %d의 최대공약수는 %d이다.", a, b, result);
    return 0;
}

```

● Problem 2 (15점)

1) 달력을 출력하는 프로그램을 작성한다.

■ 연도와 월을 입력하면 해당되는 달의 달력만 출력한다. (실행 예 참고)

■ 방법

I. 달력을 출력하기 위해서는 그 해 첫날의 요일을 알아야 한다. 다음 수식은 (year-1)년 12월 31일의 요일을 계산하기 위한 수식이다. 계산된 값이 0이면 일요일, 1이면 월요일, ..., 6이면 토요일이다. 이를 이용하여 입력한 연도의 각 날짜의 요일을 정할 수 있다.

■ $((\text{year} - 1) * 365 + (\text{year} - 1)/4 - (\text{year} - 1)/100 + (\text{year} - 1)/400) \% 7$

II. 2월은 28일과 29일인 경우가 있으며 2월이 29일인 연도는 다음 식이 참인 경우로 정할 수 있다.

■ $(!(\text{year} \% 4) \&\& (\text{year} \% 100)) || !(\text{year} \% 400)$

III. 한 주의 첫번째 요일은 일요일로 하며 오른쪽 정렬을 이용하여 줄을 맞춰서 출력이 되도록 한다.

연과 월을 입력하세요.: 2022 11

2022/11

	1	2	3	4	5
6	7	8	9	10	11
13	14	15	16	17	18
20	21	22	23	24	25
27	28	29	30	31	

```
#include <stdio.h>

// 맨 첫줄의 공백 계산하는 함수입니다.
int get_space(int total)
{
    return total % 7;
}

// (year-1)년 까지의 총 요일을 계산하는 함수입니다.
int get_totalday(int y)
{
    int y1 = y - 1;
    int total = 365 * y1;
    total += y1 / 400 - y1 / 100 + y1 / 4;

    return total;
}

//윤년을 판별하는 함수입니다.
int year_cal(int y)
{
    if (y % 400 == 0 || (y % 100 != 0 && y % 4 == 0)) return 1;
    return 0;
}
```

Problem 2, Problem 3 공통 부분에 해당하는 code

```

int main(void)
{
    int i, j, k;
    int year, month, total, spacevar;

    //1월부터 12월까지 각 월의 마지막 일을 저장합니다.
    int month_last = 0, target_last = 0;
    printf("연도를 입력하세요. ");
    scanf_s(" %d %d", &year,&month);

    printf("%d/%dWnWn", year, month);
    // (year-1)년 까지의 총 요일을 계산합니다.
    total = get_totalday(year);

    for (i = 0; i < month; i++)
    {
        // 1월, 3월, 5월, 7월, 8월, 10월, 12월에 해당됩니다.
        if ((i <= 6 && i % 2 == 0) || (i > 6 && i % 2 == 1)) month_last = 31;
        // 2월에 해당됩니다.
        else if (i == 1)
        {
            month_last = 28;
            // 윤년을 판별하는 함수를 호출합니다. 리턴 값이 1이라면 2월달에 +1을 해줍니다.
            if (year_cal(year))
                month_last += 1;
        }
        // 4월, 6월, 9월, 11월에 해당됩니다.
        else month_last = 30;

        //target_last는 입력받은 month의 마지막 날입니다.
        target_last = month_last;
        //total에 각 달의 마지막 날의 수를 더해줍니다.
        total += month_last;
    }

    // 마지막으로 더해진 달은 출력해야 하는 달이므로, 다시 total에서 빼줍니다.
    total = total - target_last;

    // 출력하는 달의 첫주의 요일 앞에 공백이 몇 개인지 계산합니다.
    spacevar = get_space(total + 1);

    // 계산한 첫 주, 첫 요일 앞 공백을 출력합니다.
    for (k = 0; k < spacevar; ++k)
        printf(" ");

    // 입력받은 달의 1일부터 마지막 일 까지 출력합니다.
    for (j = 1; j <= target_last; ++j)
    {
        printf("%3d", j);
        // (j+spacevar) % 7 == 0이라면, 한줄을 출력 완료한 것이므로 줄바꿈을 출력합니다.
        if ((j + spacevar) % 7 == 0) printf("Wn");
    }
    printf("WnWn");

    return 0;
}

```

참고 코드 : 배열을 사용한 방법

```
int main(void)
{
    int i, j, k;
    int year, month, total, spacevar;

    //1월부터 12월까지 마지막날의 일 수를 배열에 저장합니다.
    int month_last_day[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

    printf("연과 월을 입력하세요. ");
    scanf_s("%d %d", &year, &month);

    // (year-1)년 까지의 총 요일을 계산합니다.
    total = get_totalday(year);

    // 윤년을 판별하는 함수를 호출합니다. 리턴 값이 1이라면 true에 해당되므로, 2월달에 +1을 해줍니다.
    if (year_cal(year))
        month_last_day[1] += 1;

    // year년의 1월부터 (month-1)월까지 총 일 수를 total에 더합니다.
    for (i = 0; i < month - 1; i++)
    {
        total += month_last_day[i];
    }
    printf("%d/%d\n\n", year, month);

    // 출력하는 달의 첫주의 요일 앞에 공백이 몇 개인지 계산합니다.
    spacevar = get_space(total + 1);
    // 계산한 첫 주, 첫 요일 앞 공백을 출력합니다.
    for (k = 0; k < spacevar; ++k)
        printf(" ");
    // 입력받은 달의 1일부터 마지막 일 까지 출력합니다.
    for (j = 1; j <= month_last_day[i]; ++j)
    {
        printf("%3d", j);
        // (j+spacevar) % 7 == 0이라면, 한줄을 출력 완료한 것이므로 줄바꿈을 출력합니다.
        if ((j + spacevar) % 7 == 0) printf("\n");
    }
    printf("\n\n");

    return 0;
}
```

● **Problem 3 (15점)**

1) 달력을 출력하는 프로그램을 작성한다.

■ 연도를 입력하면 그 해의 달력을 모두 출력한다

■ 방법

I. Problem 2에서와 같이 그 해의 첫날의 요일과 2월의 날짜 수를 계산하고 이를 이용하여 전체 달력을 출력한다

II. 한 주의 첫번째 요일은 일요일로 하며 오른쪽 정렬을 이용하여 줄을 맞춰서 출력이 되도록 한다.

공통분에 해당하는 code 참조

```
int main(void)
{
    int i, j, k;
    int year, month, total, spacevar;

    //1월부터 12월까지 각 월의 마지막 일을 저장합니다.
    int month_last;
    printf("연도를 입력하세요. ");
    scanf_s("%d", &year);

    // (year-1)년 까지의 총 요일을 계산합니다.
    total = get_totalday(year);

    for (int i = 0; i < 12; i++)
    {
        // 1월, 3월, 5월, 7월, 8월, 10월, 12월에 해당됩니다.
        if ((i <= 6 && i % 2 == 0) || (i > 6 && i % 2 == 1)) month_last = 31;
        // 2월에 해당됩니다.
        else if (i == 1)
        {
            month_last = 28;
            // 윤년을 판별하는 함수를 호출합니다. 리턴 값이 1이라면 2월달에 +1을 해줍니다.
            if (year_cal(year))
                month_last += 1;
        }
        // 4월, 6월, 9월, 11월에 해당됩니다.
        else month_last = 30;

        printf("%d/%d\\n\\n", year, i+1);

        // 출력하는 달의 첫주의 요일 앞에 공백이 몇 개인지 계산합니다.
        spacevar = get_space(total + 1);

        // 계산한 첫 주, 첫 요일 앞 공백을 출력합니다.
        for (k = 0; k < spacevar; ++k)
            printf(" ");

        // i번째 달의 1일부터 마지막 일 까지 출력합니다.
        for (j = 1; j <= month_last; ++j)
        {
            printf("%3d", j);
            // (j+spacevar) % 7 == 0이라면, 한줄을 출력 완료한 것이므로 줄바꿈을 출력합니다.
            if ((j + spacevar) % 7 == 0) printf("\\n");
        }
        printf("\\n\\n");

        // 현재 출력하는 달의 마지막을 total에 더해줍니다. 이는 다음 반복시 공백 계산에 사용됩니다.
        total += month_last;
    }
    return 0;
}
```


참고 코드 : 배열을 사용한 방법

```
int main(void)
{
    int i, j, k;
    int year, month, total, spacevar;

    // 1월부터 12월까지 마지막날의 일 수를 배열에 저장합니다.
    int month_last_day[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

    printf("연도를 입력하세요. ");
    scanf_s("%d", &year);

    // (year-1)년 까지의 총 요일을 계산합니다.
    total = get_totalday(year);

    // 윤년을 판별하는 함수를 호출합니다. 리턴 값이 1이라면 true에 해당되므로, 2월달에 +1을 해줍니다.
    if (year_cal(year))
        month_last_day[1] += 1;;

    for (int i = 0; i < 12; i++)
    {
        printf("%d/%dWnWn", year, i+1);

        // 출력하는 달의 첫주의 요일 앞에 공백이 몇 개인지 계산합니다.
        spacevar = get_space(total + 1);

        // 계산한 첫 주, 첫 요일 앞 공백을 출력합니다.
        for (k = 0; k < spacevar; ++k)
            printf(" ");

        // i번째 달의 1일부터 마지막 일 까지 출력합니다.
        for (j = 1; j <= month_last_day[i]; ++j)
        {
            printf("%3d", j);
            // (j+spacevar) % 7 == 0이라면, 한줄을 출력 완료한 것이므로 줄바꿈을 출력합니다.
            if ((j + spacevar) % 7 == 0) printf("Wn");
        }
        printf("WnWn");

        // 현재 출력하는 달의 마지막 일을 total에 더해줍니다. 이는 다음 반복시 공백 계산에
        // 사용됩니다.
        total += month_last_day[i];
    }
    return 0;
}
```

공통 부분에 해당하는 code 참조