Ex 1: Write a program that checks if a given year is a leap year (a year that is divisible by 4, except for the cases where it is divisible by 100 and not by 400).

Some examples: 500, 700 and 900 are not leap years. 400, 800 and 1200 are, as well as 96, 124, and 704.
int year=2100; // or use scanf
if( year%4 == 0) {
    if( year%100 == 0) {
        if ( year%400 == 0) {
            printf("%d is a leap year.",
            year);
        } else {
            printf("%d is not a leap year.",
            year);
        }
    } else {
        printf("%d is a leap year.",
        year);
    }
} else {
    printf("%d is not a leap year.",
    year);
}
Ex 2: Write a program that accepts a day and a month (assume that is it correct input) and print the date of exactly a week later, in the next two cases:

a) Every month has exactly 30 days in it.

b) The number of days is the real one: (1-31, 2-28, 3-31, 4-30, 5-31, 6-30, 7-31, 8-31, 9-30,10-31,11-30, 12-31).
int day, month, new_day, next_month, new_month;
day=25; //or use scanf
month=12; //or use scanf

new_day = (((day - 1) + 7) % 30) + 1;
next_month = (((month - 1) + 1) % 12) + 1;
new_month = (new_day < 8) ? next_month : month;

printf("A week later the date will be %d/%d\n",
new_day, new_month);
int days_in_month;
switch (month) {
    case 1 : days_in_month = 31; break;
    case 2 : days_in_month = 28; break;
    case 3 : days_in_month = 31; break;
    case 4 : days_in_month = 30; break;
    case 5 : days_in_month = 31; break;
    case 6 : days_in_month = 30; break;
    case 7 : days_in_month = 31; break;
    case 8 : days_in_month = 31; break;
    case 9 : days_in_month = 30; break;
    case 10 : days_in_month = 31; break;
    case 11 : days_in_month = 30; break;
    case 12 : days_in_month = 31; break;
}
new_day = (((day - 1) + 7) % days_in_month) + 1;
next_month = (((month - 1) + 1) % 12) + 1;
new_month = (new_day < 8) ? next_month : month;
• This solution (of part b) works, however there are cases to unify, therefore switch is not the best solution here, and we could you the operator || and && (and condition operators) to make a preferable solution.
int days_in_month;
If (month==1 || month==3 || month==5 || month==7
|| month==8 || month==10 || month==12)
{
    days_in_month = 31;
}
else if (month==4 || month==6 || month==9 ||
month==11)
{
    days_in_month = 30;
}
else /* February case..*/
{
    days_in_month = 28;
}
new_day = (((day - 1) + 7) % days_in_month) + 1;
next_month = (((month - 1) + 1) % 12) + 1;
new_month = (new_day < 8) ? next_month : month;
Ex 3 We define a positive integer number as “double odd” if it has odd number of odd digits. For example, the number 12345 is double odd, as it has 3 odd digits (1,3,5), and the number 1234 is not double odd, as it has 2 odd digits (1,3).

Write a program that accepts a positive integer, and prints
The number is double odd: v
if its double odd, and
The number is double odd: x
otherwise.
Solve it using two approaches:
a) Using condition sentence.
b) Using only logical expressions (tricky...)
int val, oddCounter = 0;
scanf("%d", &val);
while (val > 0){
    int digit = val % 10;
    if (digit % 2) {
        oddCounter++;
    }
    val /= 10;
}
printf("The number is double odd: %c",
       (oddCounter % 2) ? 'v' : 'x');
```c
int val, oddCounter = 0;
scanf("%d", &val);
while (val > 0) {
    int digit = val % 10;
    if (digit % 2) {
        oddCounter++;
    }
    val /= 10;
}
printf("The number is double odd: %c",
    (oddCounter % 2) ? 'v' : 'x');
```
int val, oddCounter = 0;
int isDigitOdd; /* a Boolean flag */
scanf("%d", &val);
while (val > 0){
    int digit = val % 10;
    isDigitOdd = digit % 2;
    oddCounter += isDigitOdd;
    val /= 10;
}
printf("The number is double odd: %c",
    (oddCounter % 2) ? 'v' : 'x');
int val, oddCounter = 0;
int isDigitOdd, isCounterOdd; /* Boolean flags */
scanf("%d", &val);
while (val > 0){
    int digit = val % 10;
    isDigitOdd = digit % 2;
    oddCounter += isDigitOdd;
    val /= 10;
}

isCounterOdd = oddCounter % 2; /* 1 for odd, 0 for even */
printf("The number is double odd: %c", 'x'+isCounterOdd*('v'-x'));