次に示す「年内での経過日数を求める」プログラムを入力し、自分のパソコンでコンパイル、実行できることを確認した後、次の問に答えなさい.なお、プログラムの日本語部分は、英語、ローマ字に変更してかまいません.

```
/*年内の経過日数を求める*/
#include <stdio.h>
/*- 各月の日数 -*/
int mdays[][13] = {
  \{-1, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31\}
  \{-1, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31\}
};
/*--- 西暦 year 年のカレンダー ---*/
int *calendar(int year) {
 if (year % 4 == 0 && year % 100 != 0 || year % 400 == 0)
   return (int *)mdays + 13; /* (int *) 強制的に型変換する命令 */
   return (int *) mdays;
/*--- 西暦 y 年 m 月 d 日の年内の経過日数を求める ---*/
int dayofyear(int y, int m, int d) {
 int *i;
 int days = d;
/* 日数 */
 for (i = calendar(y) + m - 1; *i != -1; i--) {
                                           (1)
   days += *i ;
 return (days);
int main(void) {
 int year, month, day; /* 年・月・日 */
                        /* もう一度? */
 int retry;
 do {
   printf("年:"); scanf("%d", &year);
   printf("月:"); scanf("%d", &month);
   printf("日:"); scanf("%d", &day);
   printf("年内で%d 日目です。\Yn", dayofyear(year, month, day));
   printf("もう一度しますか(1…はい/0…いいえ):");
   scanf ("%d", &retry);
 \} while (retry == 1);
 return (0);
```

間(1) 西暦で年を指定すると、その年のカレンダーへのポインターを返す関数 calendar を用いて、次のような式を書いた、このときの式の値がいくらになるかを求めなさい。

*calendar(2019)	(*calendar(2019)) +2	*(calendar(2019) + 2)

- 問(2) 関数 dayofyear を「dayofyear(2019, 3, 1)」として呼び出した.
 - (ア) このとき, 関数 dayofyear の for ループ内の「days += *i」は何回実行されますか.
 - (イ) 関数 dayofyear の for 文のループが始まる直前の下線①の位置での*i の値はいくらですか.
 - (ウ) 関数 dayofyear の for 文が終了したとき, i が指すポインターの位置は, 配列 mdays の何処を指していますか. mdays[0][0]のように, 配列 mdays の添え字を用いて示してください.
- 問(3) 各月の日数を記憶する配列 mdays の初期値を以下のように変更した. この変更した初期値でも「年内での経過日数を正しく求める」ように、関数 *calendar と関数 dayofyear を変更したプログラム(ソースファイル名は kadai2.c とすること)を作成してください.

```
/*- 各月の日数 -*/
int mdays[][13] = {
    {0, 31, 28, 31, 30, 31, 30, 31, 30, 31, 30, 31},
    {5, 31, 29, 31, 30, 31, 30, 31, 30, 31, 30, 31}
};
```