Designing a Computer Program

Programming is easy. It's knowing what to program that is difficult.

Flowcharts and Programming

- Outline a programming task with a flowchart.
- Relate the flowchart to programming structures.
- Create a flowchart during this presentation.

Top Down Design

- Start with a large task.
- Use a flowchart to outline the task.
- Break the task into smaller more understandable tasks.
- Divide the smaller tasks into subtasks if it’s easier to understand.

Top Down Design

- Clearly state the problem you are trying to solve.
- Define the inputs for the problem.
- Define the required output.
- Visualize the procedure with a flowchart.

Problem:
Count the number of student birthdays in each month and on what day

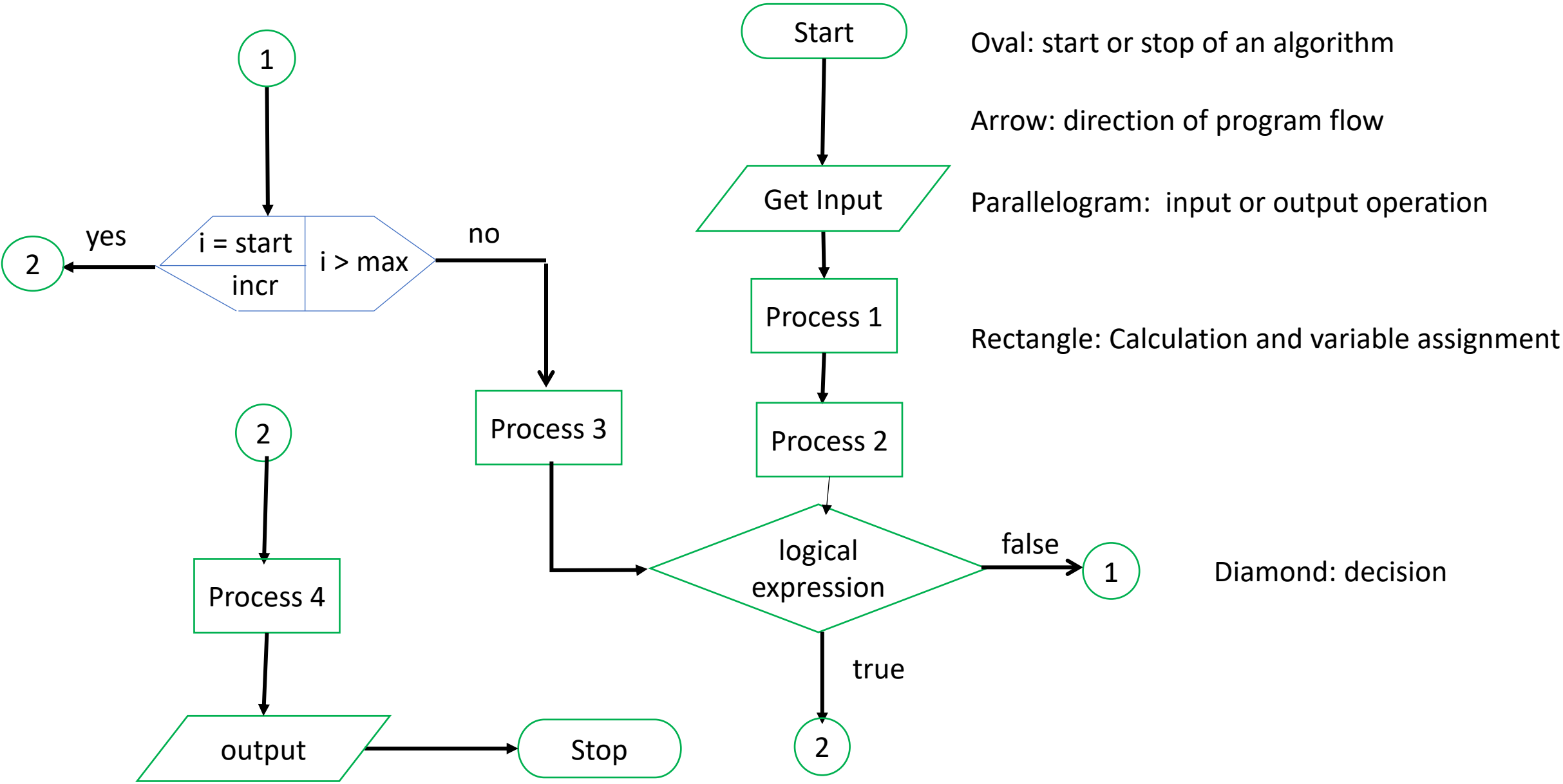
Necessary Inputs

1. How many students are there?

Output Format

1. Number in each month?
2. Number on each day?
3. Or the number on each day of the month?

Flowchart symbols



Oval: start or stop of an algorithm

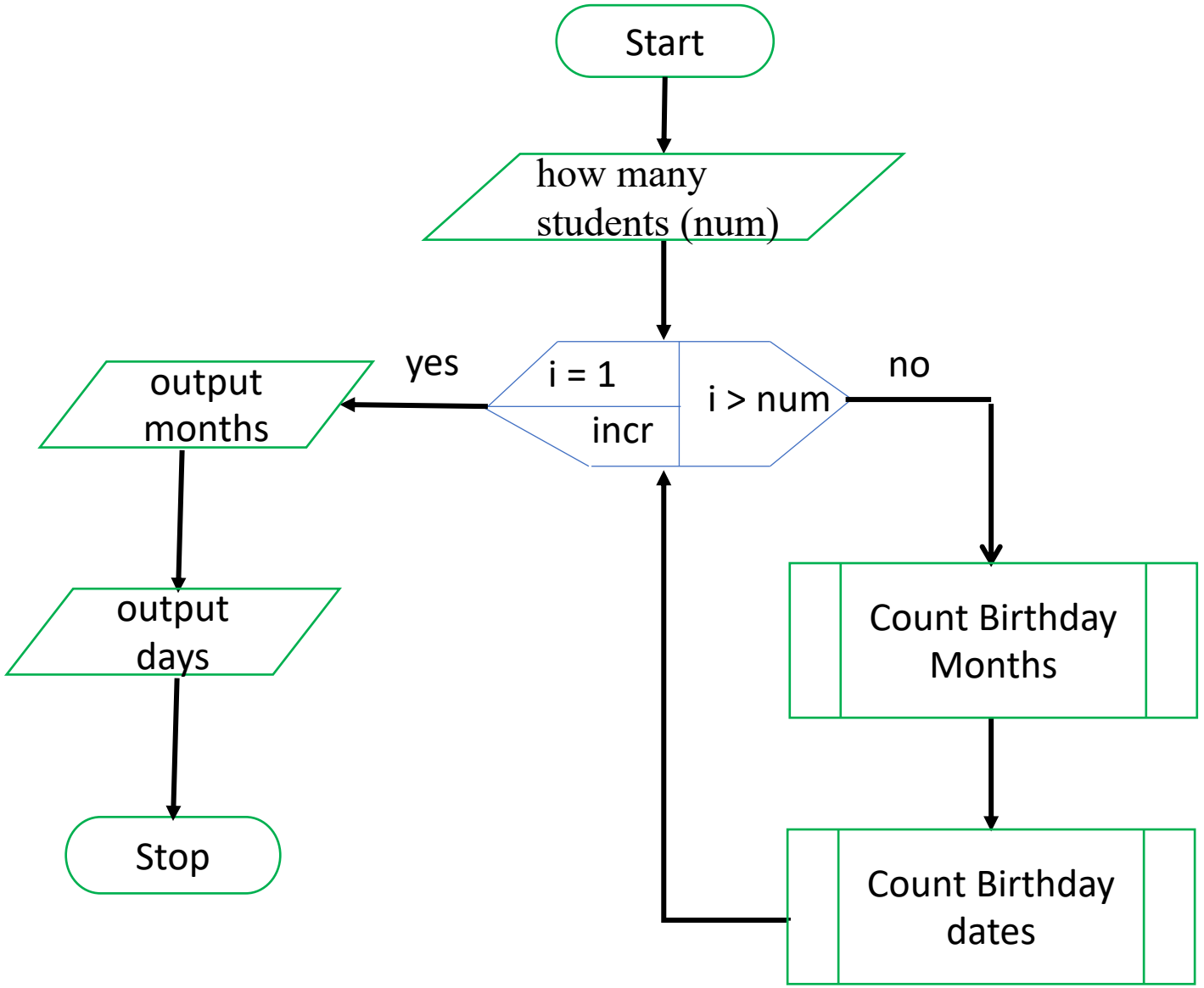
Arrow: direction of program flow

Parallelogram: input or output operation

Rectangle: Calculation and variable assignment

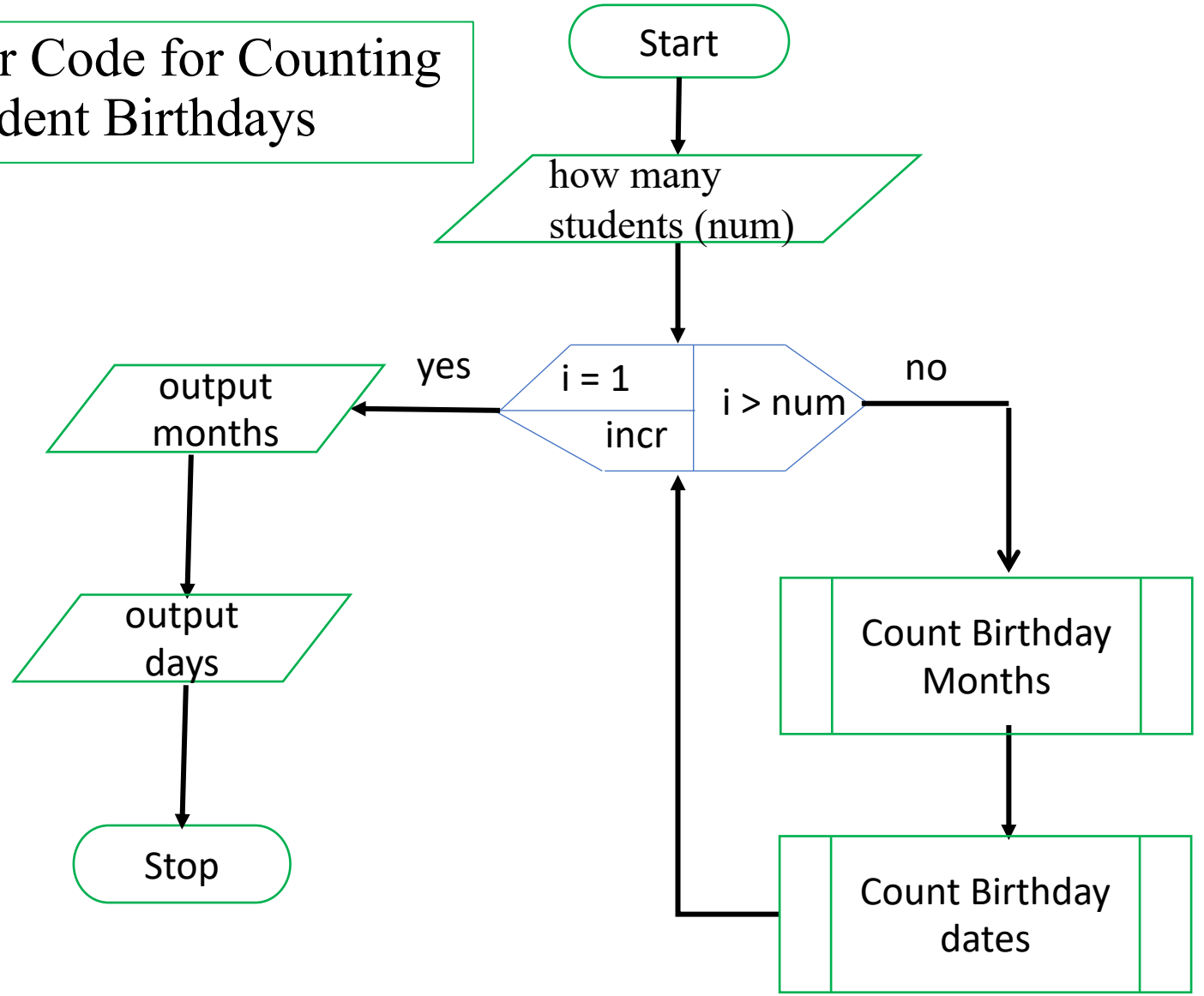
Diamond: decision

Flowchart for Counting Birthdays

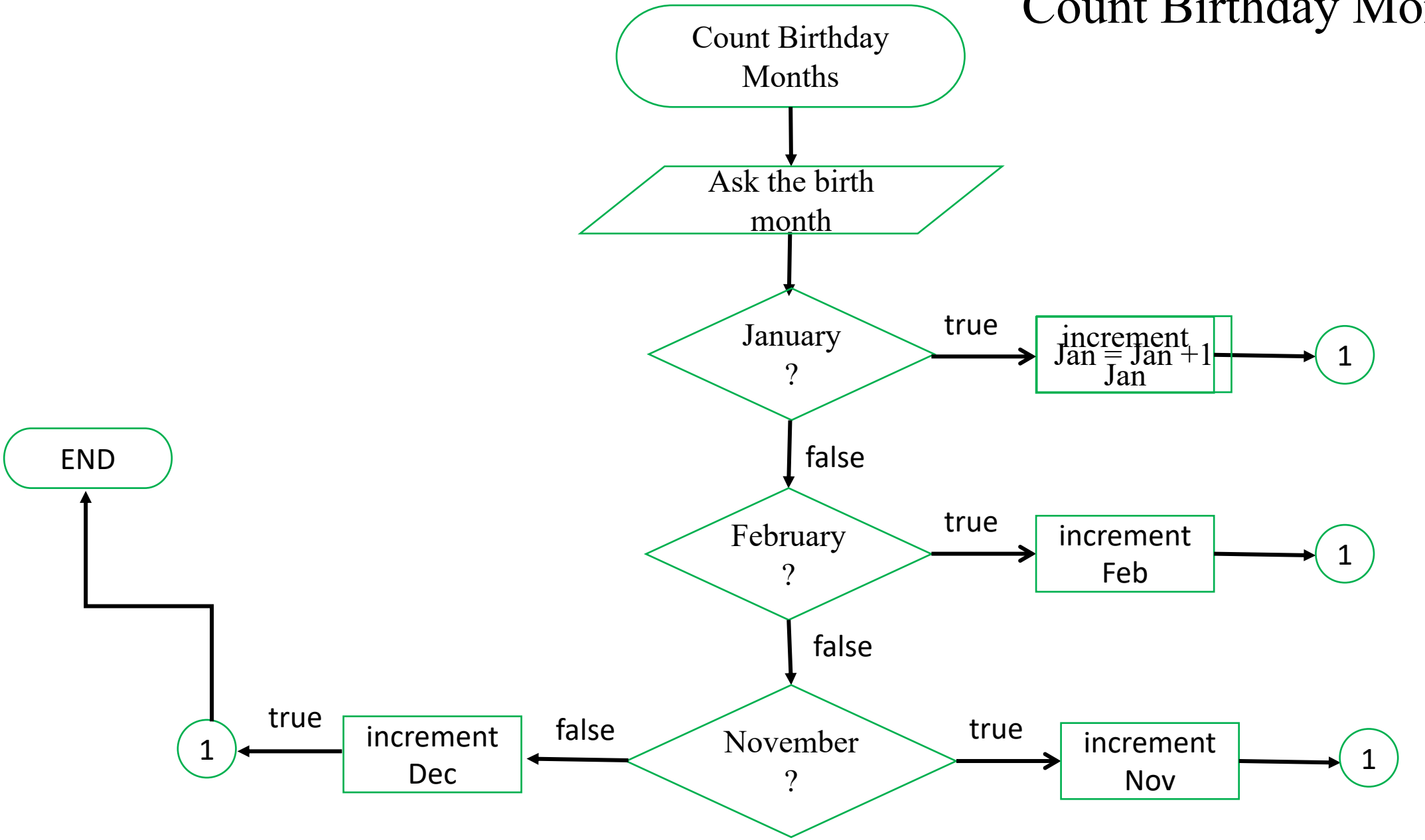


Computer Code for Counting Student Birthdays

```
Program Count_Birthday_dates
! This program counts the number of students born in
!  each month and the number of students born on each day of
!  the month.

Read (*,*) num          ! the number of students to be queried
Read (*,*) month,       ! the birthday month
Read (*,*) day          ! the weekday          ! each student

i = 1                   ! initialize the student counter
Do While (i .LE. num)
        Call    Count_Months (month)
        Call    Count_days (day)
        i = i + 1       !  i=i+1 difficult to read
End While

Write (*.*)  jan, feb, …, dec
Write(*,*) i1, i2, i3, … i31
END Program Count_Birthday_dates
```

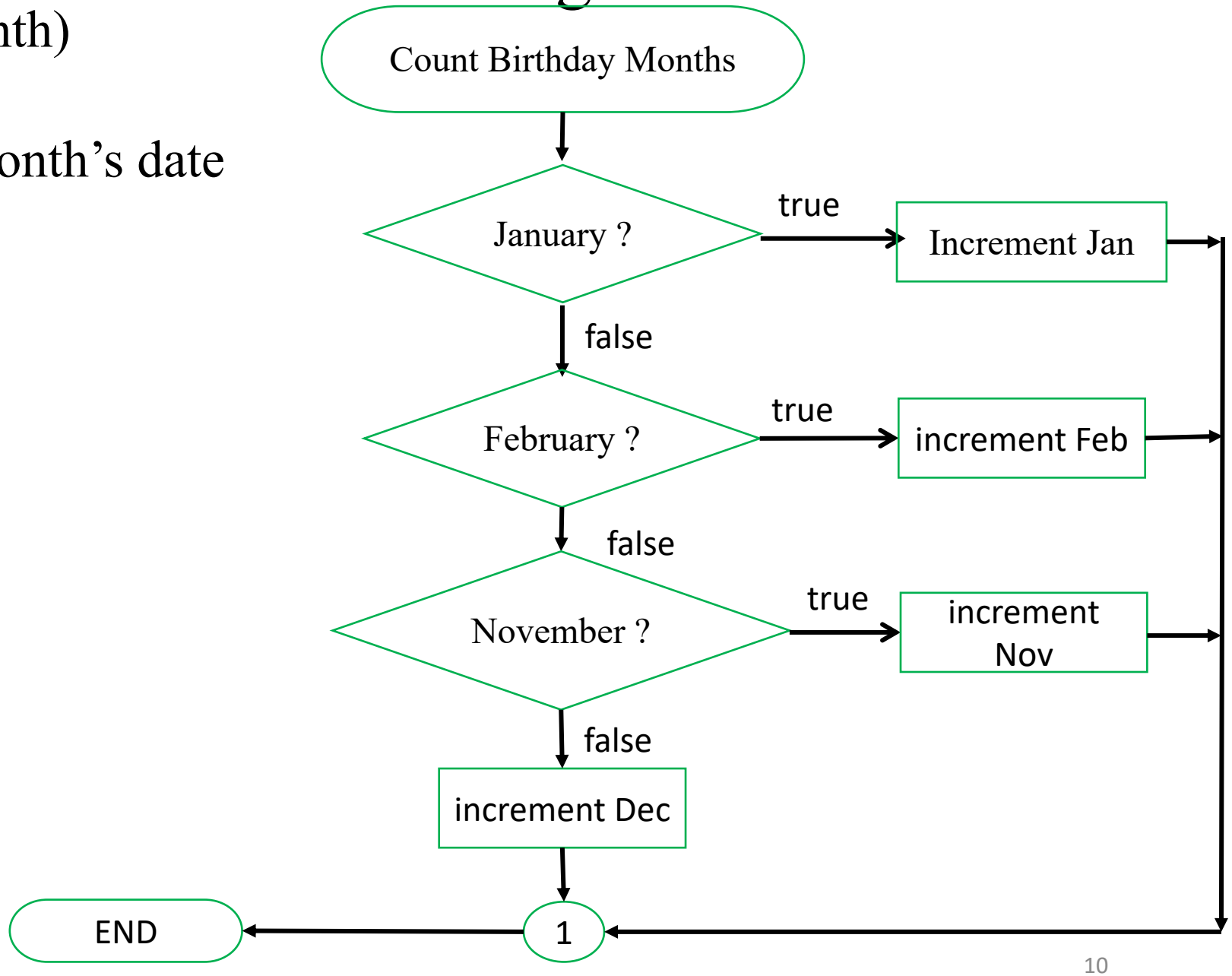


Count Birthday Months



Computer Code for Counting Months

```
Subroutine Count_month (month)

! month is an integer for the month's date
! January = 1, March =3, etc.

    if (month = 1) then
        Jan = Jan + 1
    elseif (month = 2)
        Feb = Feb + 1
    elseif  (month = 11)
        nov = nov + 1
    else
        dec = dec + 1
    endif
End Program Count_month
```



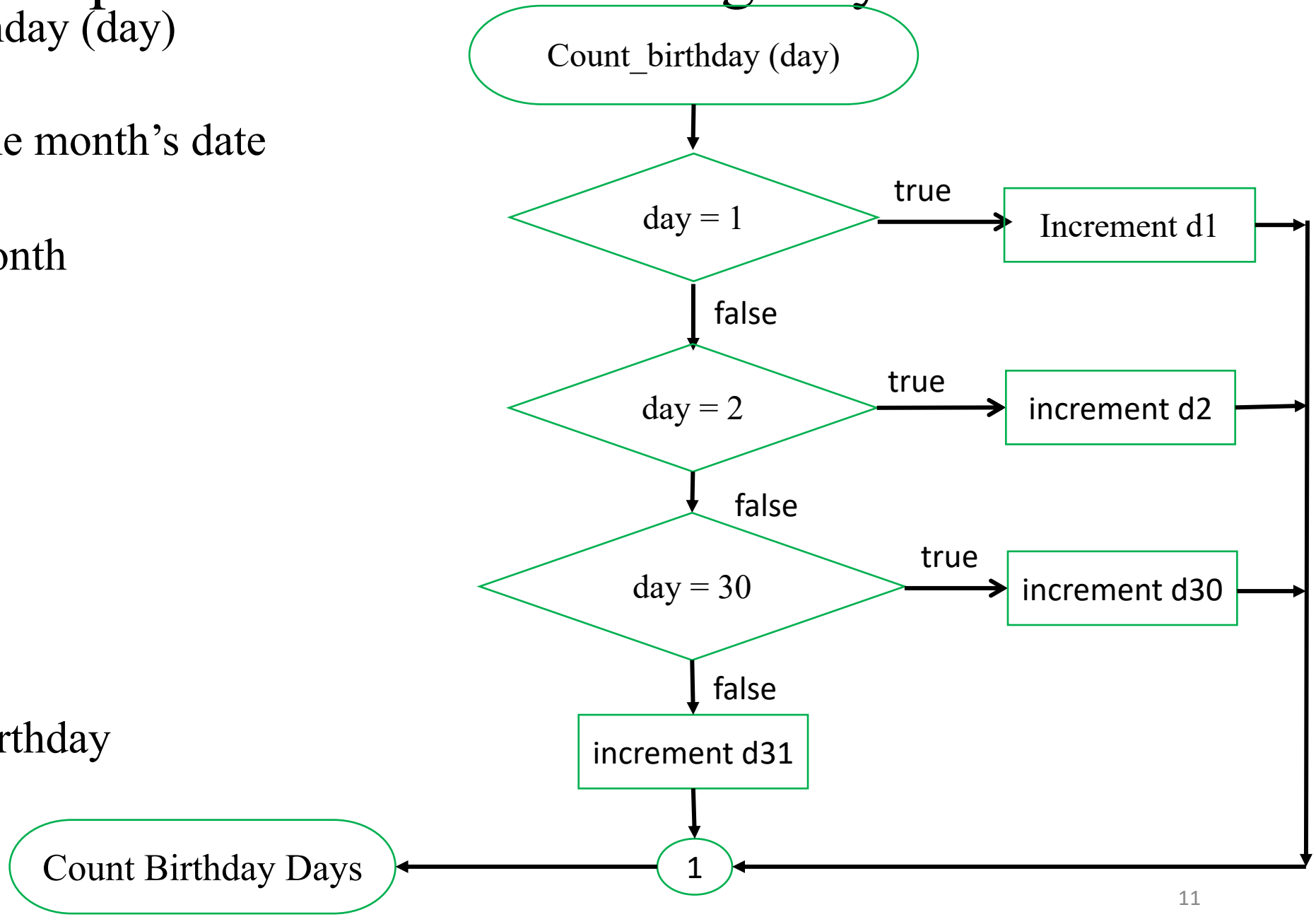
Computer Code for Counting Days

Subroutine Count_Birthday (day)

! day is an integer for the month's date
! d1 = 1st, d2 =2nd, etc.
! d# is the day of the month

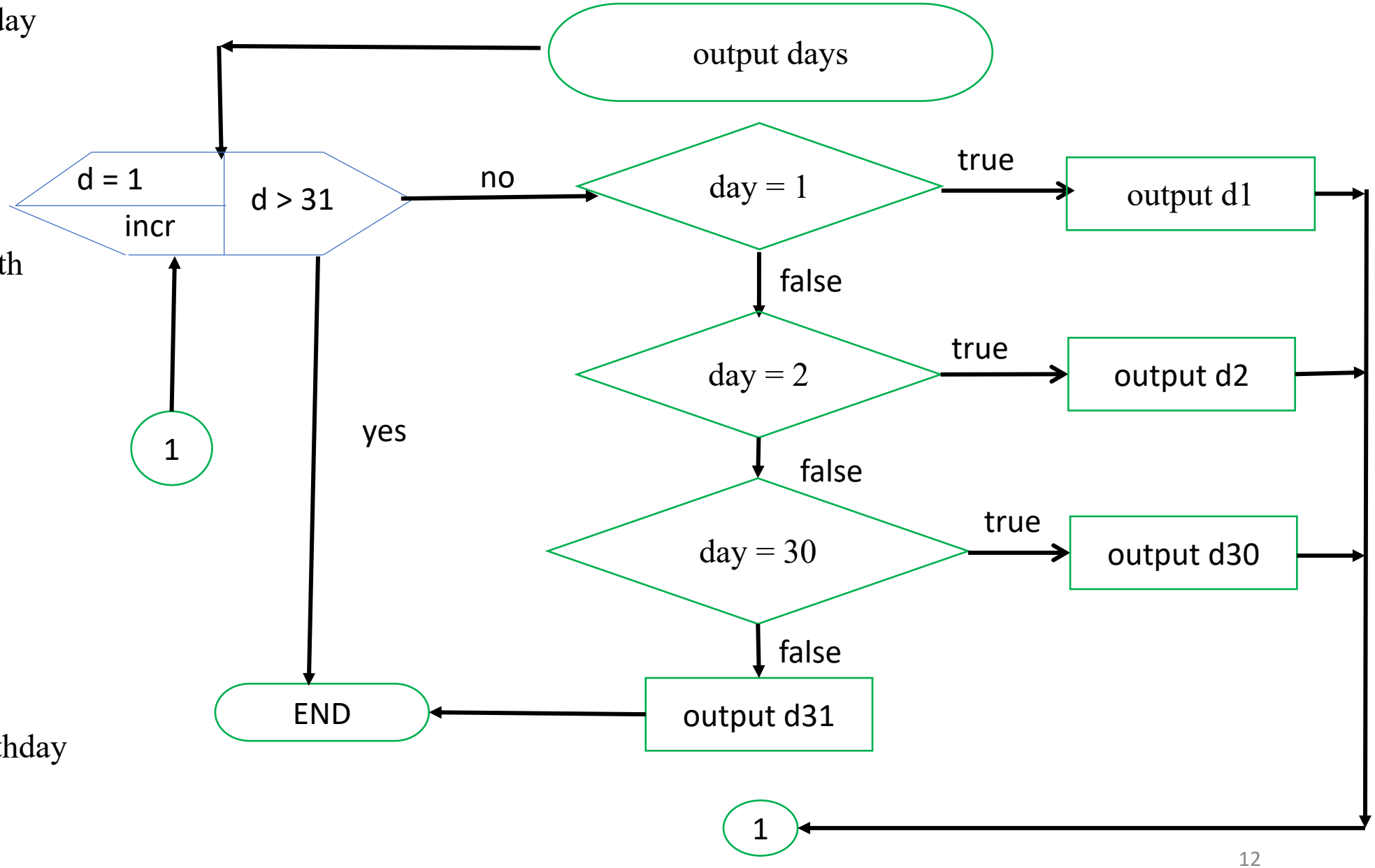
```
if (day = 1) then
      d1 = d1 + 1
elseif (day = 2)
      d2 = d2 + 1
elseif  (day = 30)
      d30 = d30 + 1
else
      d31 = d31 + 1
endif
```

End Program Count_Birthday



Computer Code for Outputing days

```
Subroutine Output_Birthday
(day)

! day is an integer for the
!month's date
! d1 = 1st, d2 =2nd, etc.
! d# is the day of the month
day = 1
do while day .LE. 31
        if (day = 1) then
                write d1
        elseif (day = 2)
                write d2
        elseif  (day = 30)
                write d30
        else
                write d31
        endif
endwhile
End Program Output_Birthday
```



Computer Code for Outputing Months

```
Subroutine output_month (month)

! month is an integer for the month's number
month = 1
Do while month < .LE. 12
        if (month = 1) then
                output jan
        elseif (month = 2)
                output Feb
        elseif  (month = 11)
                output nov
        else
                output dec
        endif
endwhile
End Program output_month
```