Examples Using if

Example 21 (contd)

```
#include <stdio.h>
int main() {
    int y1, m1, d1, y2, m2, d2, same, earlier;

    printf("Enter first date (mm/dd/yy): ");
    scanf("%d/%d/%d", &m1, &d1, &y1);
    printf("Enter second date (mm/dd/yy): ");
    scanf("%d/%d/%d", &m2, &d2, &y2);

    same = y1 == y2 && m1 == m2 && d1 == d2;
    earlier = (y1 < y2) || (y1 == y2 && m1 < m1) ||
              (y1 == y2 && m1 == m2 && d1 < d2);

    // Rest of the program
}
```

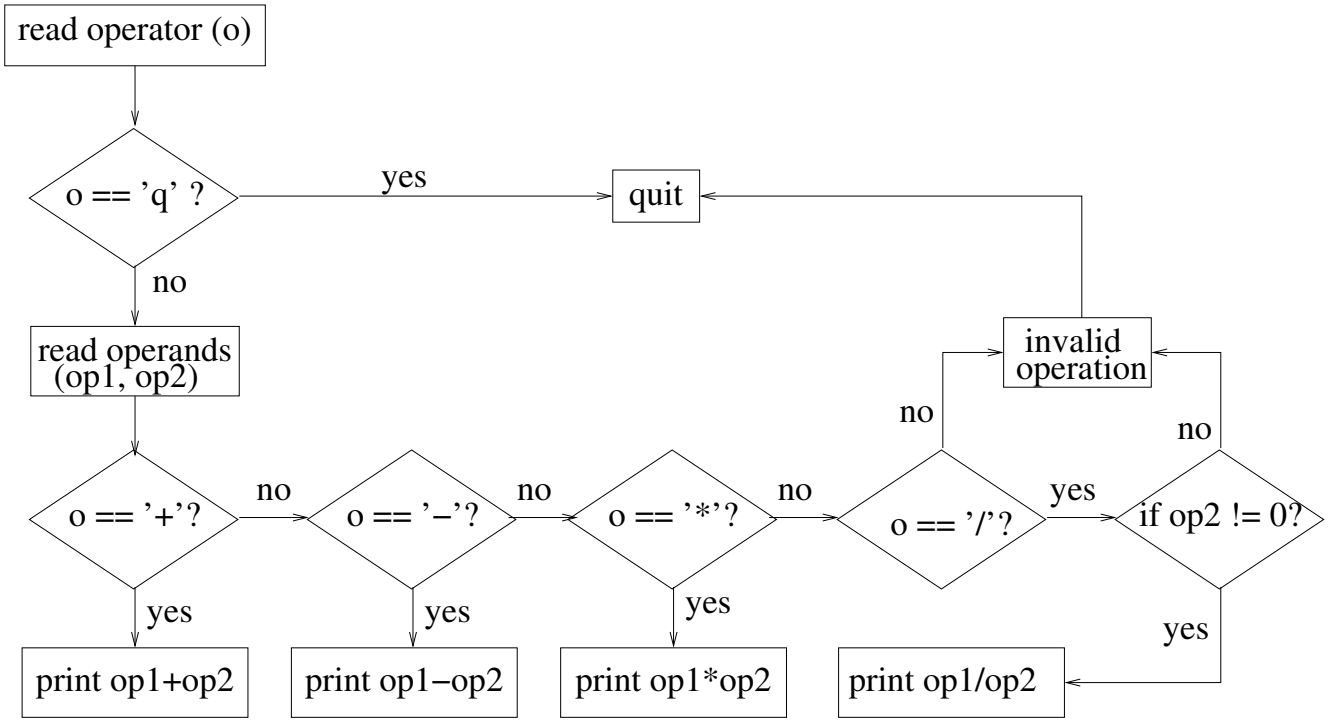
Examples Using if

Example 21 (contd)

```
if (earlier)
        printf("%d/%d/%d precedes %d/%d/%d\n",m1,d1,y1,m2,d2,y2);
else if (same)
        printf("Dates are same\n");
    else
        printf("%d/%d/%d precedes %d/%d/%d\n",m2,d2,y2,m1,d1,y1);
```

Examples Using if

Example 22



Examples Using if

Example 22 (contd)

```
#include <stdio.h>

int main() {
    char o;
    int op1, op2;

    printf("Enter operator(+,-,*,/), q to quit : ");
    scanf("%c", &o);
    if (o == 'q') return 0;
    else {
        printf("Enter operands : ");
        scanf("%d %d", &op1, &op2);
    }
    // Plug the rest part of the program below this line
}
```

Examples Using if

Example 22 (contd)

```
if (o == '+')
    printf("%d␣%c␣%d␣=␣%d\n", op1, o, op2, op1 + op2);
else if (o == '-')
        printf("%d␣%c␣%d␣=␣%d\n", op1, o, op2, op1 - op2);
    else if (o == '*')
            printf("%d␣%c␣%d␣=␣%d\n", op1, o, op2, op1 * op2);
        else if (o == '/' && op2 !=0)
                printf("%d␣%c␣%d␣=␣%d\n", op1, o, op2, op1 / op2);
            else
                printf("invalid␣operation\n");
```

Dangling else

- When nested **if**s are used, the problem of dangling **else** should be addressed.
- For example, consider

```
if (testscore >= 50)
    if (testscore > 90)
        printf("You got an A grade\n");
else // testscore is between 50 to 90
        printf("You failed\n");   // WRONG
```

Indentation imply **else** is meant to be paired with **outer if**, but it pairs with **inner if**.

Dangling else

Solution to Dangling Else

Enclose inner **if** by braces as shown in order to avoid **dangling else**.

```
if (testscore >= 50) {
    if (testscore > 90)
        printf("You got an A grade\n");
    else
        // testscore is between 50 to 90
} else
    // testscore is less 50.
    printf("You failed\n");
```

It is safer to enclose the statement(s) within if block by braces to avoid dangling else.

Conditional Operator

Conditional Operator

Sometime instead of **if else**, the conditional operator **? :** could be used to perform one of the two actions.

expression 1 ? expresssion 2 : expression 3

```
int i, j, k;
i = 1;
j = 2;
k = i > j ? i : j;          // k = 2 here
k = (i >= 0 ? i : 0) + j; // k = 3 here
```

Conditional Operator

Conditional Operator

Another simple effective use of conditional operator is:

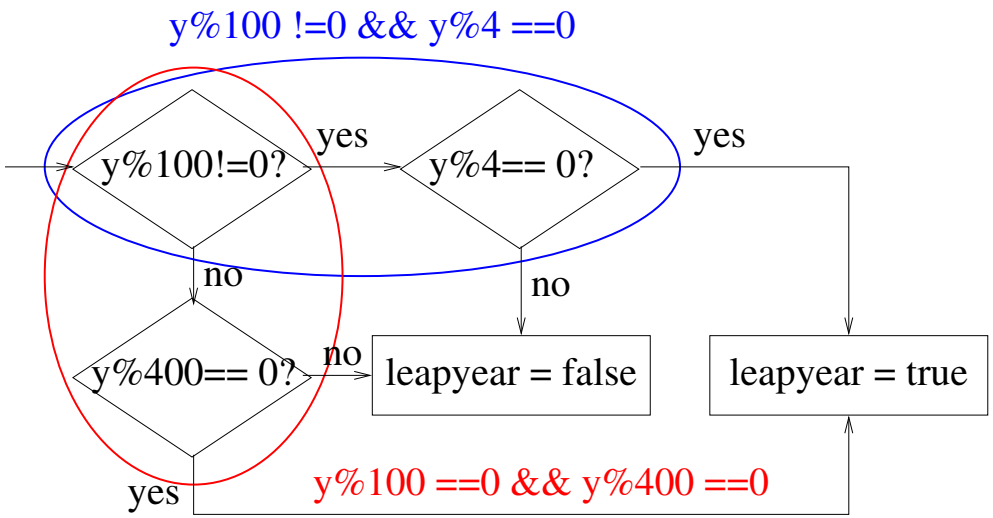
```
#include <stdio.h>
#define MAX(X, Y) X > Y ? X : Y

int main() {
    int i = 18, j = 57;
    printf("Max is: %d\n", MAX(i, j));
}
```

Conditional Operator

Conditional Operator

Similarly we can define macro for leap year:




```
#define LEAP(Y) ((Y%100 != 0) && (Y%4 == 0) || (Y% 400 == 0) ?  1 :  0;
```

Conditional Operator

Example 23

```
#include <stdio.h>
#define LEAP(Y) ((Y%100 != 0) && (Y%4 == 0)) || (Y%400 == 0)? 1 : 0
int main() {
    int d, m, y, nd;

    printf("Enter month (1-12): ");
    scanf("%d", &m);
    printf("Enter year : ");
    scanf("%d", &y);

    nd = (m == 4 || m == 6 || m == 9 || m == 11)
    if (m != 2)
        if (nd) d = 30;
        else d = 31;
    else if (LEAP(y)) d = 29;
         else d = 28;
    printf("No. of days in month %d of year %d is: %d\n", m, y, d);
}
```

Switch-Case & Break

Multiple Alternatives

- **Switch-case** is used for multiple alternatives.
- For example, in case of mini calculator, 4 alternatives exists of operator.
- Instead of writing deep nested **if else** it is natural to use **switch-case**.
- **Switch-case**, however needs **break**.
- **Break** is a jump outside the **switch-case** block to sequentially next statement.

Switch-Case & Break

Syntax