# Class 4 Control Flow (1)

How to make a decision in C Program?

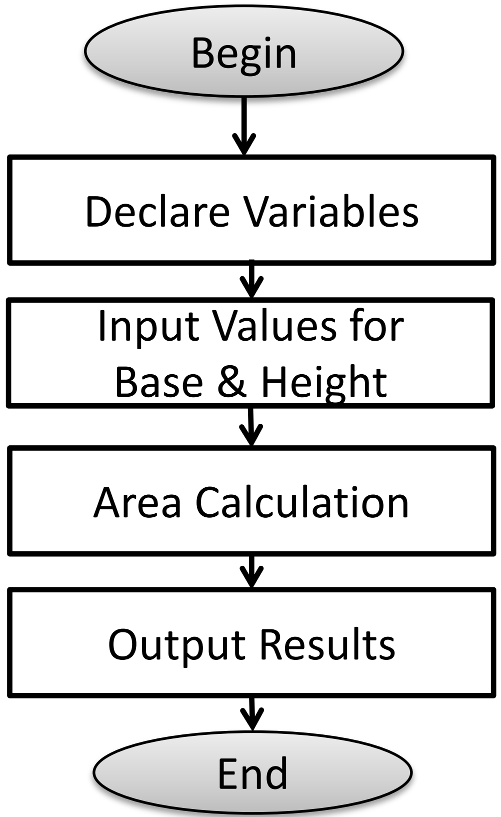
## How does a programmer control the C program?

- A programmer can control the execution of a C program by using three kinds of control structures
  - Sequence Structure (顺序结构)
  - Selection Structure (选择结构)
  - Repetition Structure (循环结构)

# Sequence Structure

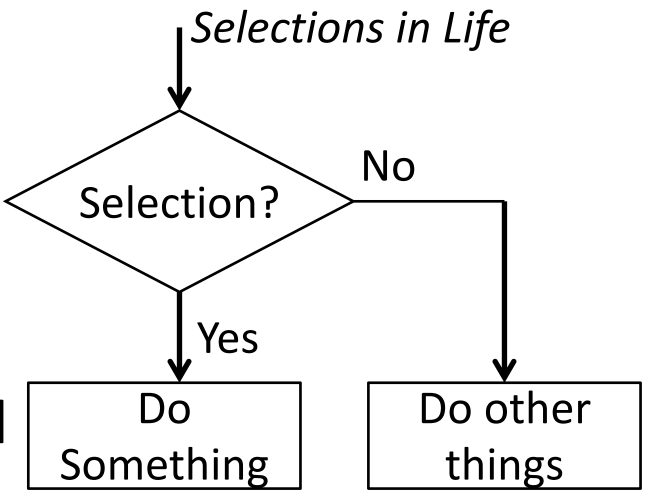
- The program is executed sequentially (line by line)
- An Example
  - Given the base and height of a triangle, calculate its area and output the result

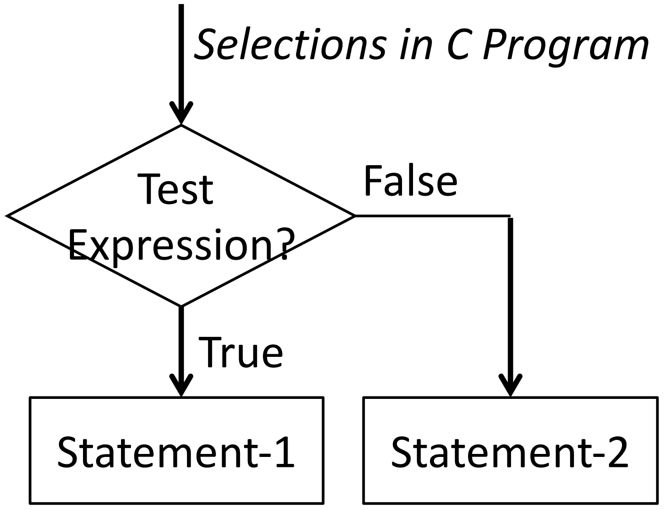
```
# include <stdio.h>

int main()
{
    float base, height, area;
    scanf(“%f %f”, &base, &height);
    area = base * height / 2.0;
    printf(“base = %.2f\n”, base);
    printf(“height = %.2f\n”), height);
    printf(“area = %.2f\n”, area);
}
```



# Selection Structure

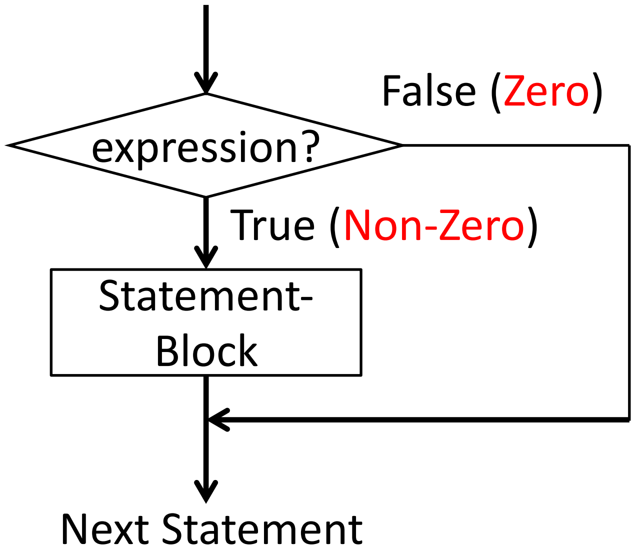
- Making Decision
  - if (bank balance is zero) borrow money
  - if (room is dark)  turn on lights
  - if (age is more than 60) person is retired
- C language possesses such decision-making capabilities by supporting the following statements
  - if / if-else statement
  - switch statement
  - Conditional operator statement
  - goto statement





# IF Statement:  A Single Branch

```
if (expression)
{
    statement-block;
}
```

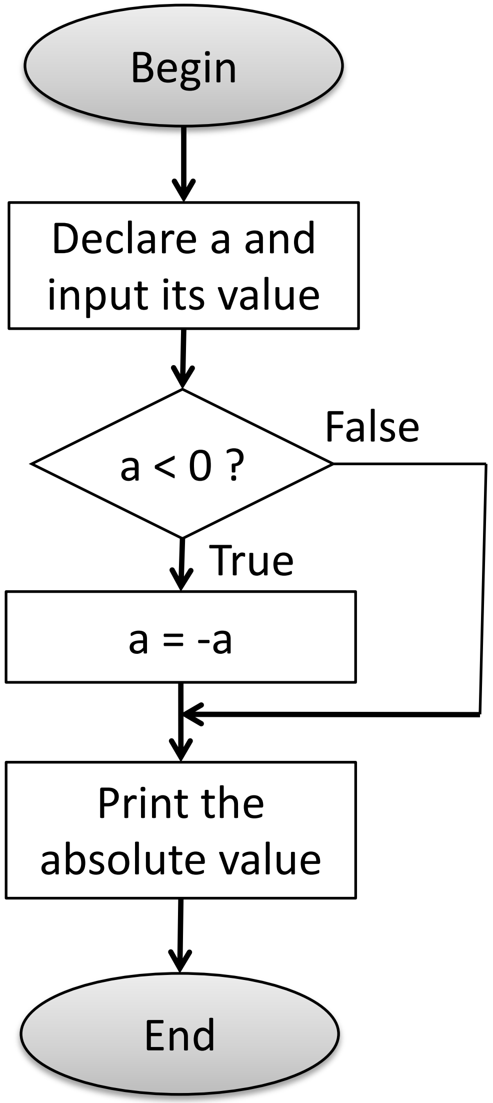


- The expression of evaluated. If it is true (a non-zero ), the statement-blocks is executed; if it is false (zero), the statement- block is ignored and the next statement is executed.

# An Example of IF Statement

- Input an integer and calculate its absolute value

```c
# include <stdio.h>

int main( )
{
    int a;
    scanf("%d",&a);
    if ( a < 0 )
    {
        a  = -a ;
    }
    printf("|a|=%d\n", a);
    return 0;
}
```



# Notes for IF Statement

- If there is only a single statement in the statement-block, the brace can be omitted

```
if ( a < 0 )
{
      a   = -a ;
}
```

```
if ( a < 0 )
      a   = -a ;
```

OR

```
if ( a < 0 )   a   = -a ;
```

- DO NOT add semicolon after the parentheses of the IF statement.

```
if ( a < 0 );
      a   = -a ;
```

```
if ( a < 0 );
{
      a   = -a ;
}
```

ERROR

This is equal to execute a None-statement when the expression of IF statement is true

# Notes for IF Statement

- Relational or logical operators that having Boolean values
  - Examples: Expression to test a leap year

- Arithmetic expressions of which the value stand for true
  - Examples:

```
if ( a - b )  printf("%d != %d", a, b );
if ( num % 2) printf(“odd number");
```

- Assignment expressions can also be tested

```
if (sum = a + b) printf("Summation is non-zero");
```

- DO NOT mis-use the two operators :
  = (assignment operator) and == (equality operator)

```
if (disc == 0) printf(“Two equal roots\n"); // CORRECT

if (disc = 0)  printf(“Two equal roots\n"); // ERROR
```

## Exercise

- A program reads 4 values a, b, c, d from the keyboard and evaluates the ratio of (a+b) to (c-d) and prints the result, if (c-d) is no equal to zero.

Method-1

```
#include <stdio.h>
int main()
{
    int a, b, c, d;
    float ratio;
    // Read four integer values
    printf( “Enter four integer values\n” );
    scanf( “%d %d %d %d” , &a, &b, &c, &d);
    if ( c-d != 0 ) // Is this test correct?
    {
         //Is type casting necessary?
        ratio = (float)(a+b) / (float)(c-d);
         printf( “Ratio =%f\n” , ratio);
    }
    return 0;
}
```

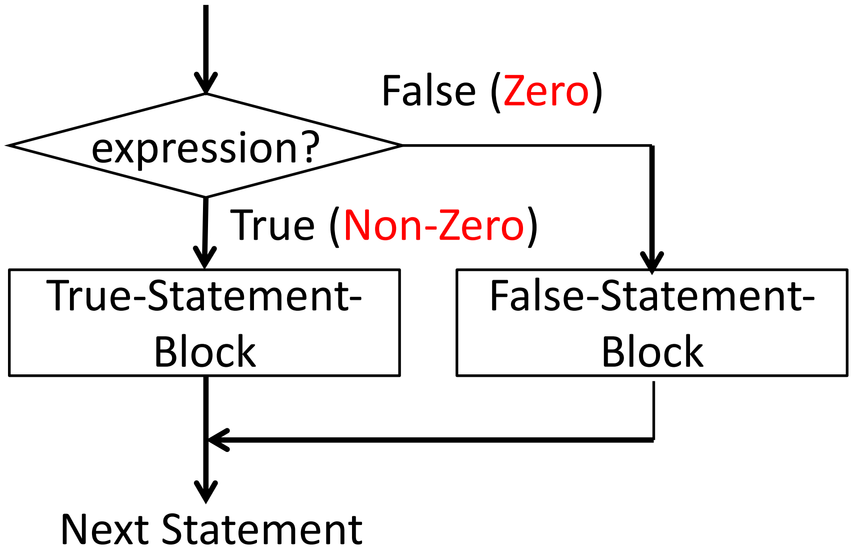
Method-2

```
#include <stdio.h>
#include <math.h>

int main()
{
    float a, b, c, d, ratio;

    printf( “Enter four floating numbers\n” );
    scanf( “%f %f %f %f” , &a, &b, &c, &d);
    if ( fabs(c-d) > 1e-6 )
    {
        ratio = (a+b) / (c-d);
        printf( “ratio =%f\n” , ratio);
    }
    return 0;
}
```

## IF-ELSE Statement:  Double Branches

```
if (expression)
{
    True-statement-block;
}
else
{
    False-Statement-block;
}
```

Flow-chart of IF-ELSE Statement

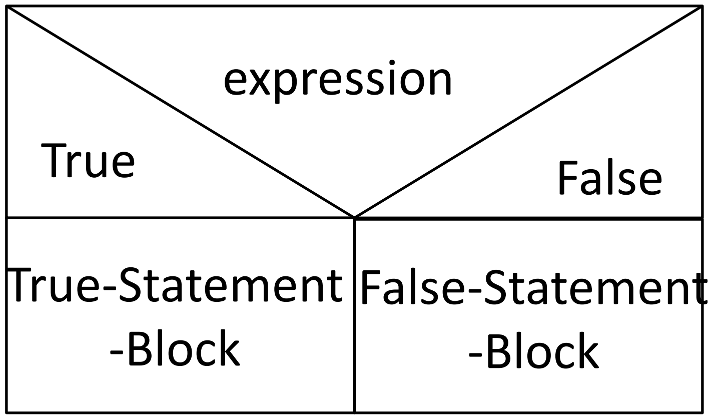


- The expression of evaluated. If it is true (a non-zero ), the statement-block-1 is executed; if it is false (zero), the statement- block-2 is executed.

# IF-ELSE Statement:  Double Branches

```
if (expression)
{
    True-statement-block;
}
else
{
    False-Statement-block;
}
```

Another kind of Flow-chart of IF-ELSE Statement

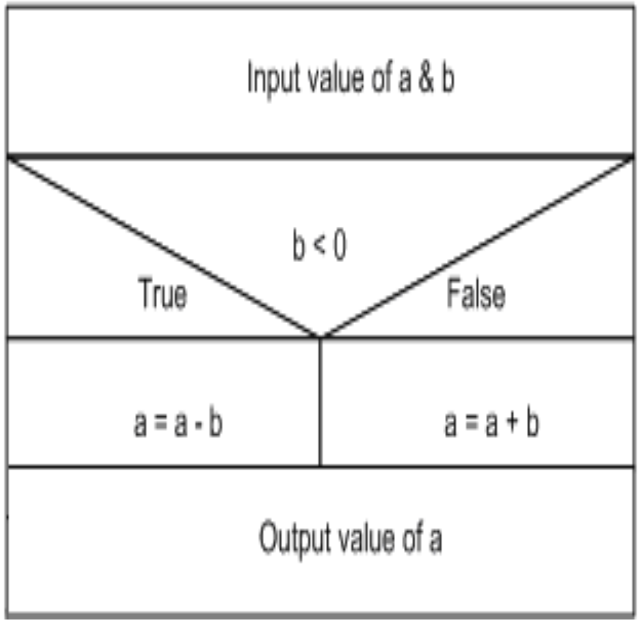


- The expression of evaluated. If it is true (a non-zero ), the statement-block-1 is executed; if it is false (zero), the statement- block-2 is executed.

# An Example of IF-ELSE Statement

- Input two integers ( a,  b) and calculate the value of  a + |b|

```
# include <stdio.h>

int main( )
{
    int a, b;
    scanf("%d %d", &a, &b);

    if ( b < 0 )
    {    a  -=  b ;    }
    else
    {    a  +=  b ;    }

    printf("a + | b |= %d\n", a) ;
    return 0;
}
```



## Notes of IF-ELSE Statement

- IF-ELSE should be used by pairs, and ELSE can not be used as a single statement.
- If there is a single statement in the True-statement-block / False-statement-block, the braces can be omitted.

```
if ( b < 0 )
    a  -=  b ;
else
    a  +=  b ;
```

# Notes of IF-ELSE Statement

- IF-ELSE statement can be replaced by two IF statements. However two IF statements are less efficient.

```
if ( b < 0 )
    a  -=  b ;
else
    a  +=  b ;
```

Logically Equal But Less Efficient

```
if ( b < 0 )
    a  -=  b ;
if ( b >= 0 )
    a  +=  b ;
```

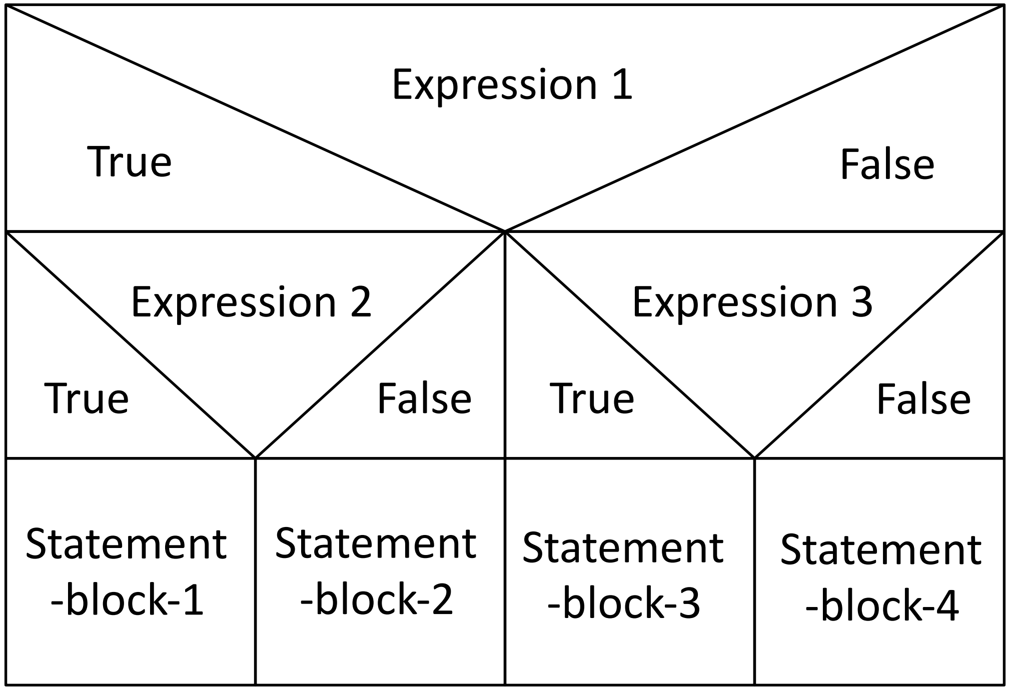
- Sometimes, IF-ELSE statement can be replaced by a conditional operator

```
a = ( b < 0 ) ? ( a – b ) : ( a + b )
```

# Nested IF-ELSE Statement

```
if (expression 1)
{
     if (expression 2)
     {
          statement-block 1;
     }
     else
     {
          statement-block 2;
     }
}
else
{
     if (expression 3)
     {
          statement-block 3;
     }
     else
     {
          statement-block 4;
     }
}
```

Flow Chart of Nested IF-ELSE Statement



# Nested IF-ELSE Statement

```
if (expression 1)
{
    if (expression 2)
    {
        statement-block 1;
    }
    else
    {
        statement-block 2;
    }
}
else
{
    if (expression 3)
    {
        statement-block 3;
    }
    else
    {
        statement-block 4;
    }
}
```

⇔

Fully-Nested IF-ELSE statement is equal to four IF statements

```
if (expression 1 && expression 2)
{
    statement-block 1;
}

if (expression 1 && !expression 2)
{
    statement-block 2;
}

if (!expression 1 && expression 3)
{
    statement-block 3;
}

if (!expression 1 && !expression 3)
{
    statement-block 4;
}
```

# Dangling else problem

- else is always paired with the most recent unpaired if.

```
if( )
if( ) statment1
else
if( ) statement2
else statement3
```

```
if( )
{
    if( ) statment1
    else
    {
        if( ) statement2
        else statement3
    }
}
```

DO NOT write codes without indents and braces

```
int x=20;
if(x>=0)
if(x<50)
printf( " okey! \n" );
else
printf( " not ok! \n" );
```

```
int x=20;
if(x>=0)
{
    if(x<50)
        printf( " okey! \n" );
    else
        printf( " not ok! \n" );
}
```

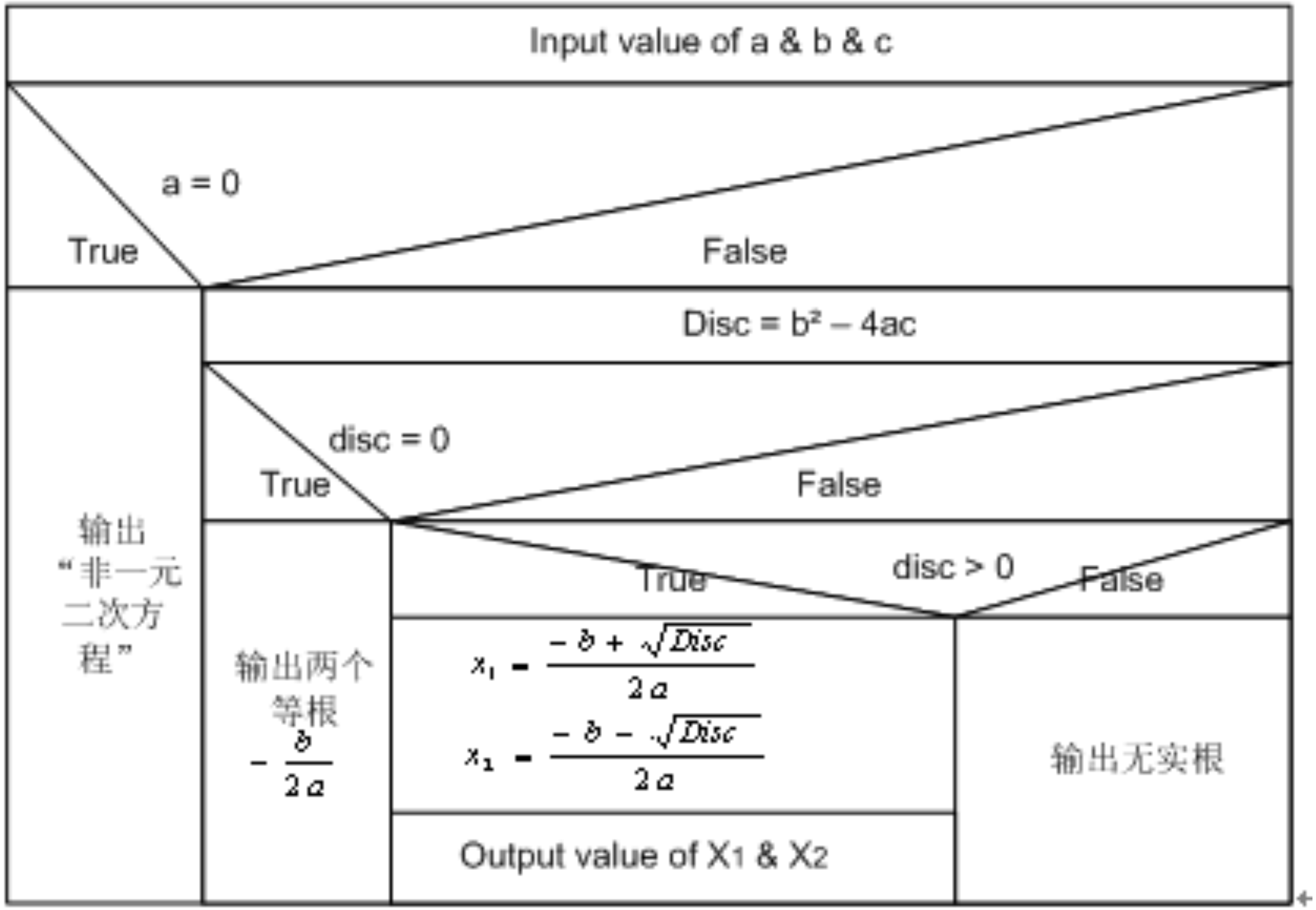
# An Example of Nested IF-ELSE Statement

- Refine your program to solve a quadratic equation

$$ax^2 + bx + c = 0$$

- Conditions:
  - a equals 0 → a linear equation
  - discriminant equals 0 → Two equal real roots
  - discriminant is greater than 0 → Two distinct real roots
  - discriminant is less than 0 → Two conjugate roots

# Flow chart

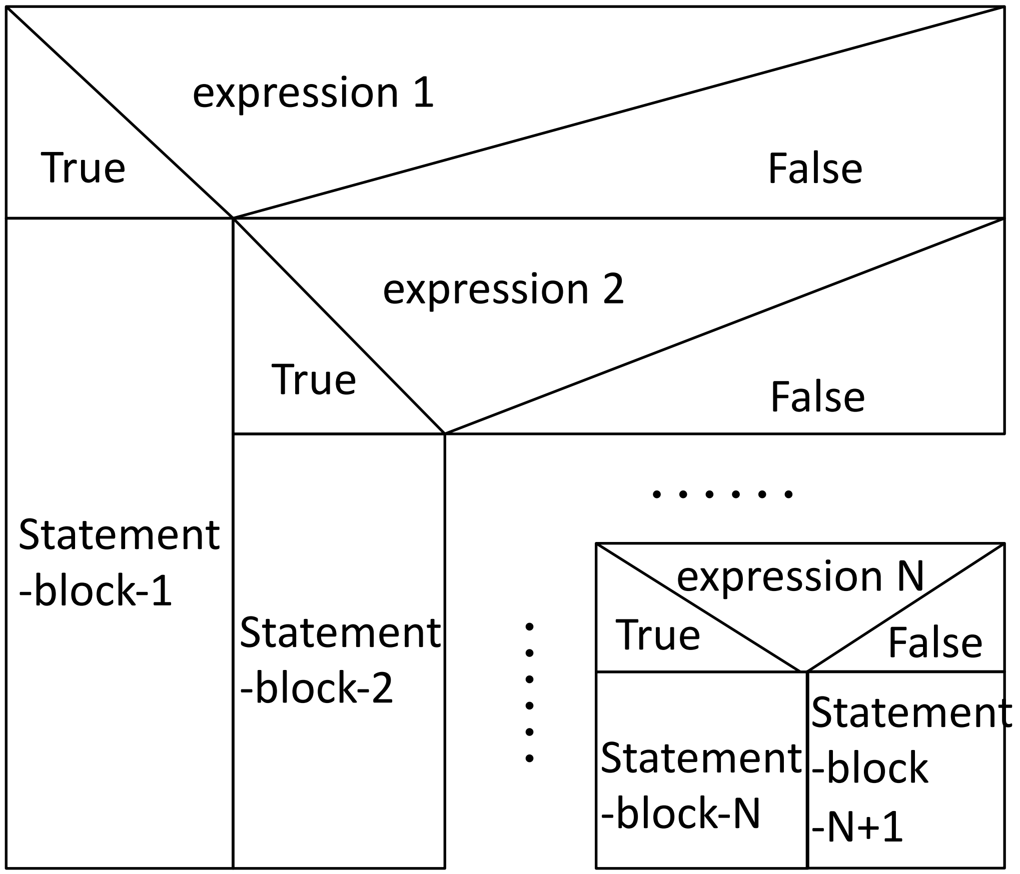

Input value of a & b & c
a = 0
True
False
输出“非一元二次方程”
Disc = b² – 4ac
disc = 0
True
False
输出两个等根 $-\frac{b}{2a}$
disc > 0
True
False
$x_1 = \frac{-b + \sqrt{Disc}}{2a}$
$x_2 = \frac{-b - \sqrt{Disc}}{2a}$
Output value of X1 & X2
输出无实根

```
#include<stdio.h>
#include<math.h>

int main()
{
    double a,b,c,disc; /*Declare variables */
    double x1,x2;

    printf("Input coefficients of the equation:\n");
    scanf(“%lf %lf %lf",&a,&b,&c);

    if(fabs(a)<=1e-8)  /*Linear equation*/
         printf("Not a quadratic");
    else         /*Quadratic equation*/
    {
         disc=b*b-4*a*c;
         ......
    }
    ......
    else    /*Quadratic equation*/
    {
         disc=b*b-4*a*c;
         if(fabs(disc)<=1e-8) /* Two equal roots*/
              printf("Two equal roots:%8.4f\n",-b/(2.0*a));
         else
         {
              if(disc>0)  /*Two distinct roots*/
              {
                x1=(-b+sqrt(disc))/(2.0*a);
                x2=(-b-sqrt(disc))/(2.0*a);
               printf("Distinct real roots:%8.4f and %8.4f\n”,x1,x2);
              }
              else /*Complex roots/
                 printf("No real roots\n");
         }
    }

    return 0;
}
```

# IF-ELSEIF-ELSE Statement (Ladder)

- IF-ELSEIF-ELSE Statement is used for multi-way decision

```
if (expression 1)
{
    statement-block 1;
}
else if (expression 2)
{
    statement-block 2;
}
......
else if (expression N)
{
    statement-block N;
}
else
{
    statement-block N+1;
}
```



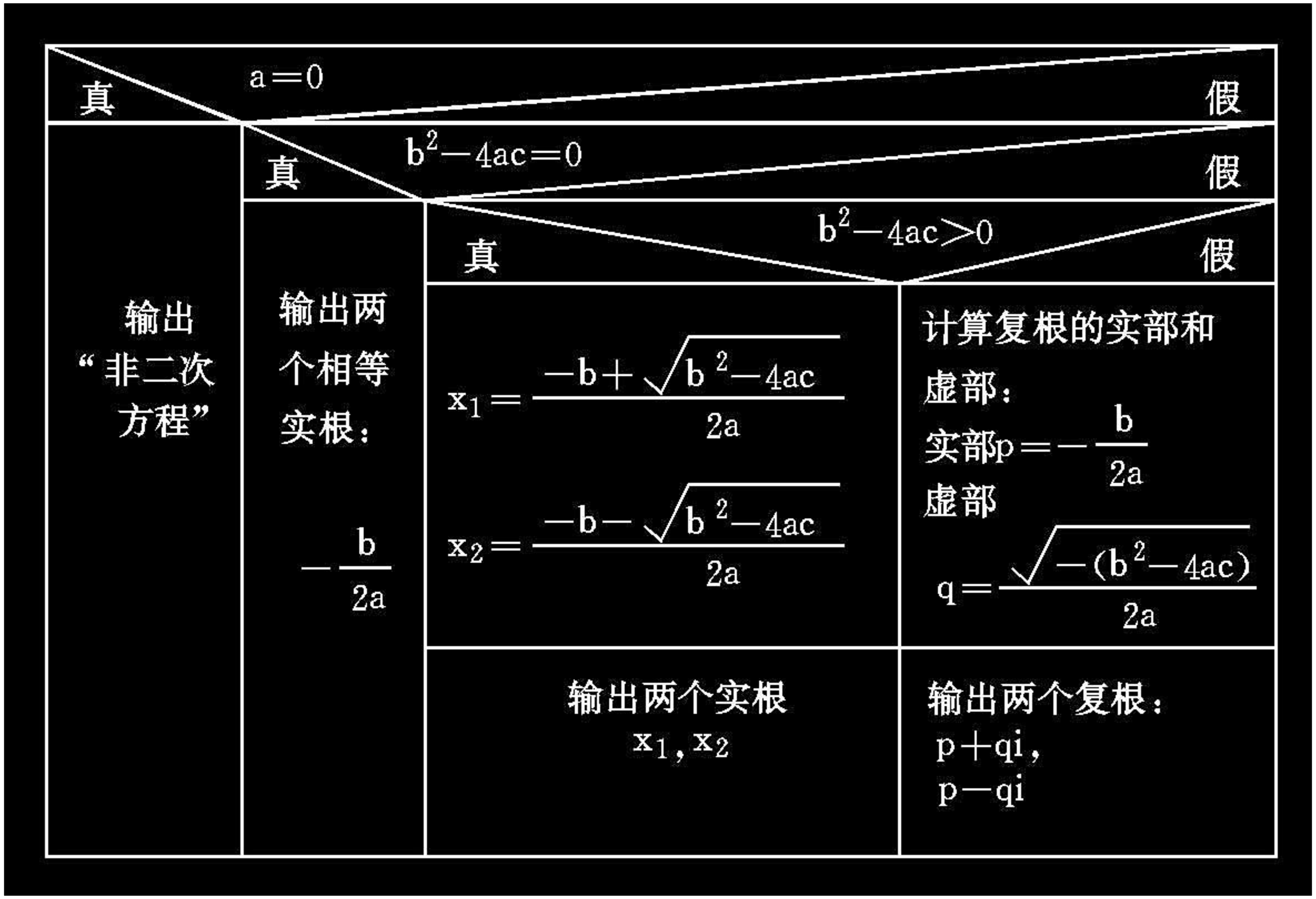
## An Example of IF-ELSEIF-ELSE Statement

- Given a score, calculate the corresponding grade and print the results according to the following rules
  - score >= 90,    grade = A
  - 80 <= score < 90, grade = B
  - 70 <= score < 80, grade = C
  - 60 <= score < 70, grade =  D
  - score < 60, grade = F

```c
#include <stdio.h>

int main()
{
    int score;
    char grade;
    scanf(“%d”, &score);

    if (score >= 90)
        grade = ‘A’;
    else if (score >= 80)
        grade = ‘B’;
    else if (score >= 70)
        grade = ‘C’;
    else if (score >= 60)
        grade = ‘D’;
    else
        grade = ‘F’;

    printf(“score = %d, grade = %c”, score, grade);

    return 0;
}
```

# Exercise

- Refine the program for quadratic equations by using IF-ELSEIF-ELSE statement.

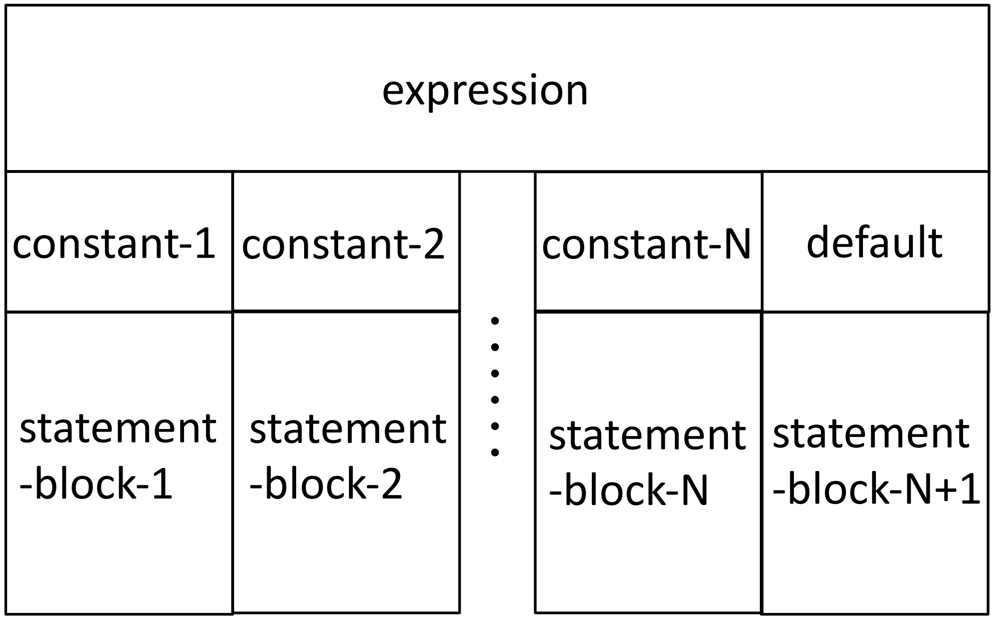


$a=0$
真
假
输出“非二次方程”
$b^2-4ac=0$
真
假
输出两个相等实根：$-\frac{b}{2a}$
$b^2-4ac>0$
真
假
$x_1=\frac{-b+\sqrt{b^2-4ac}}{2a}$
$x_2=\frac{-b-\sqrt{b^2-4ac}}{2a}$
计算复根的实部和虚部：
实部$p=-\frac{b}{2a}$
虚部
$q=\frac{\sqrt{-(b^2-4ac)}}{2a}$
输出两个实根 $x_1, x_2$
输出两个复根：$p+qi$, $p-qi$

```c
#include <stdio.h>
#include <math.h>

int main()
{
    double a,b,c,disc;
    double x1,x2;
    double realPart, imagPart;

    printf("Input coefficients of the equation:\n");
    scanf("%lf %lf %lf", &a, &b, &c);

    disc=b*b-4*a*c;

    if(fabs(a)<=1e-8)
        printf("Not a quadratic");
    else if (fabs(disc) <= 1e-8)
        printf("Two equal roots:%8.4f\n",-b/(2.0*a));
    else if (disc > 0)
    {
       ......
    }
    ......

    else if (disc > 0)
    {
        x1=(-b+sqrt(disc))/(2.0*a);
        x2=(-b-sqrt(disc))/(2.0*a);
        printf("Distinct real roots:%8.4f and %8.4f\n",x1,x2);

    }
    else
    {
        realPart = -b/(2.0*a);
        imagPart = sqrt(-disc)/(2.0*a);
        printf("Two complex roots\n");
        printf("%8.4f + %8.4fi\n", realPart, imagPart);
        printf("%8.4f - %8.4fi\n", realPart, imagPart);
    }

    return 0;
}
```

# Switch Statement

```
switch (expression)
{
    case constant-1:
        statement-block-1;
        break;
    case constant-2:
        statement-block-2;
        break;
    ......
    case constant-N:
        statement-block-N;
        break;
    default:
        statement-block-N+1;
}
```



## Notes for Switch Statement

- The switch expression must be an integral type including int, char and enum
- Case labels must be constants or constant expressions.
- Case labels must be unique and end with a colon(:).
- The break statement transfers the control out of the switch statement. Set break upon demand.
- Several case labels can share the same statement block.

```
#include <stdio.h>

int main()
{
        enum Weather{Sunny, Cloudy, Rainy};
        enum Weather today = Cloudy;

        switch(today)
        {
                case Sunny:
                        printf("T-shirt + cap\n");
                        break;
                case Cloudy:
                        printf("T-shirt + outer wear\n");
                        break;
                case Rainy:
                        printf("Raincoat + umbrella\n");
                        break;
                default:
                        printf("whatever\n");
        }

        return 0;
}
```

value of enum constant:
Sunny = 0
Cloudy = 1
Rainy = 2

```
#include <stdio.h>

int main()
{
        char a;
        scanf("%c", &a);

        switch(a)
        {
                case 'a': case 'i' : case 'u' :
                case 'e' : case 'o':
                        printf("It is a lowercase vowel letter\n");
                        break;
                case 'A': case 'I' : case 'U' :
                case 'E' : case 'O':
                        printf("It is a uppercase vowel letter\n");
                        break;
                default:
                        printf("It is not a vowel letter\n");
        }

        return 0;
}
```

These case labels share the same statement block

## Exercise

- Given a grade, use switch statement to print the corresponding range of the scores according to the following rules
  - A : score >= 90
  - B : 80 <= score < 90
  - C : 70 <= score < 80
  - D : 60 <= score < 70
  - F : score < 60

```c
#include <stdio.h>

int main()
{
    char grade;
    printf("Input the grade :\n");
    grade = getchar();
    switch(grade)
    {
        case 'A': case 'a':
            printf("score >= 90\n"); break;
        case 'B': case 'b':
            printf("80 <= score < 90\n"); break;
        case 'C': case 'c':
            printf("70 <= score < 80\n"); break;
        case 'D': case 'd':
            printf("60 <= score < 70\n"); break;
        case 'F': case 'f':
            printf("score < 60\n"); break;
        default:
            printf("Invalid grade\n");
    }
    return 0;
}
```