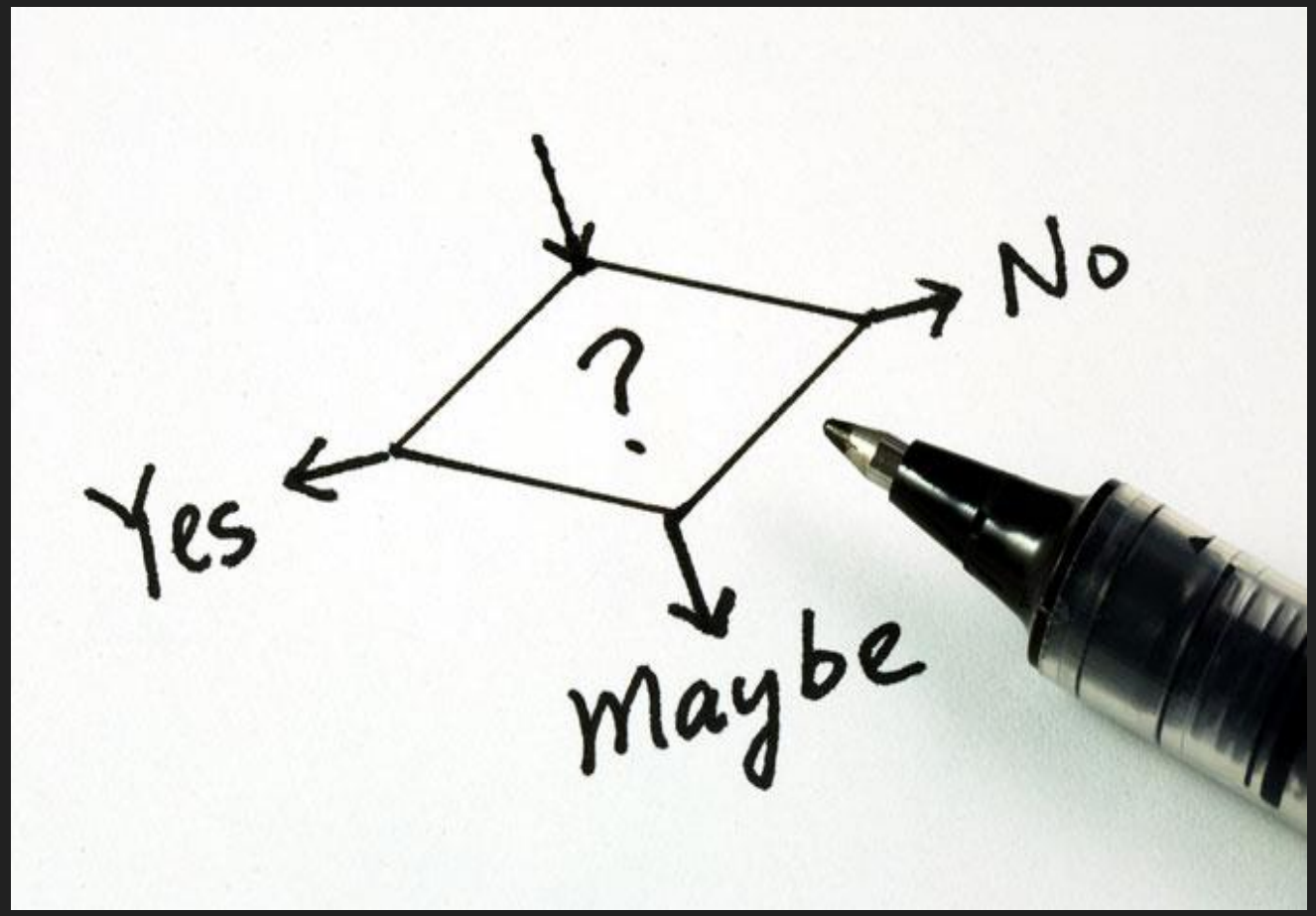
Decision making in C

Need of decision making




```
if number is odd
{
    /* code */
}
```

```
else number is even
{
    /* code */
}
```

Decision Making or Conditional Statement

- C program statements are executed sequentially.
- Decision Making statements are used to control the flow of program.
- It allows us to control whether a program segment is executed or not.
- It evaluates condition or logical expression first and based on its result (either true or false), the control is transferred to particular statement.
- If result is true then it takes one path else it takes another path.

Decision Making Statements in C

Decision Making Statements are

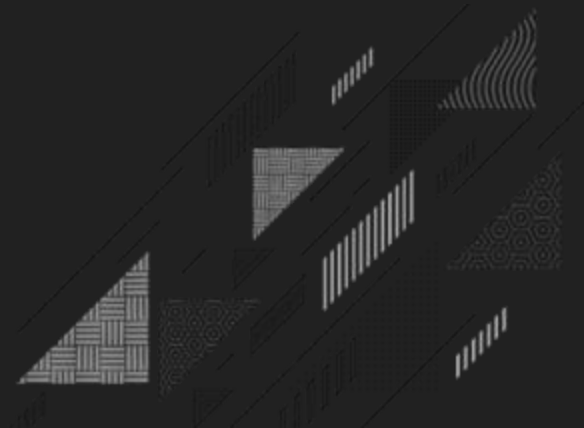
One way Decision:	`if`	(Also known as simple if)
Two way Decision:	`if…else`	
Multi way Decision:	`if…else if…else if…else`	
Two way Decision:	`?:` (Conditional Operator)	
n-way Decision:	`switch…case`	

Relational Operators

- Relational Operator is used to compare two expressions.
- It gives result either true or false based on relationship of two expressions.

Math	C	Meaning	Example	Result
>	>	is greater than	`5 > 4`	`true`
≥	>=	is greater than or equal to	`5 >= 4`	`true`
<	<	is less than	`5 < 4`	`false`
≤	<=	is less than or equal to	`5 <= 4`	`false`
≠	!=	is not equal to	`5 != 4`	`true`
=	==	is equal to	`5 == 4`	`false`

If statement



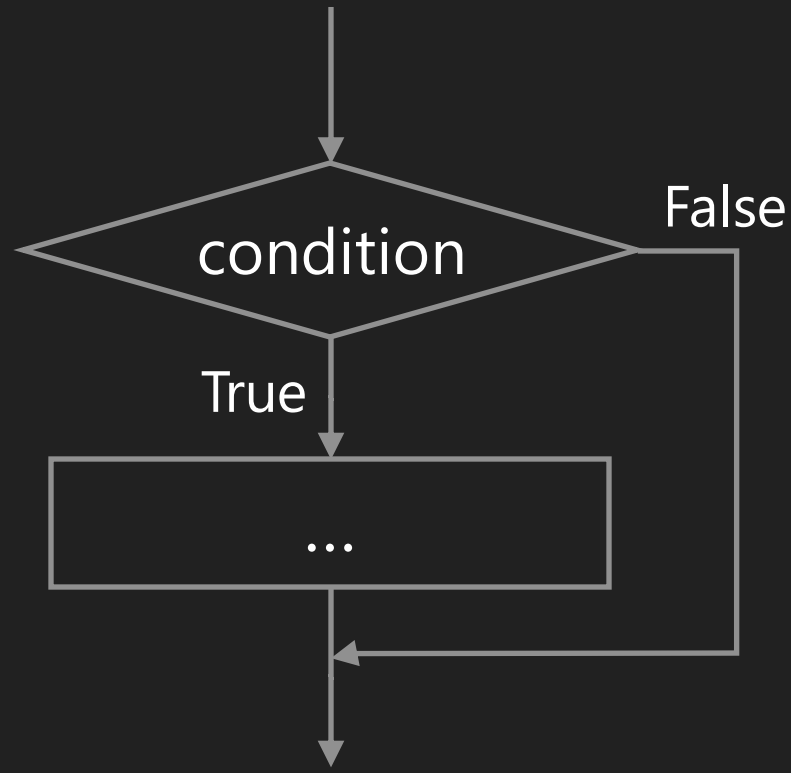
if

- `if` is single branch decision making statement.
- If condition is `true` then only body will be executed.
- `if` is a keyword.

Syntax

```
if(condition)
{
    // Body of the if
    // true part
}
```

Flowchart of `if`



WAP to print Zero if given number is 0

Program

```c
#include<stdio.h>
void main()
{
    int a;
    printf("Enter Number:");
    scanf("%d",&a);
    if(a == 0)
    {
        printf("Zero");
    }
}
```

Output

```
Enter Number:0
Zero
```

WAP to print Positive or Negative Number

Program

```c
#include<stdio.h>
void main()
{
    int a;
    printf("Enter Number:");
    scanf("%d",&a);
    if(a >= 0)
    {
        printf("Positive Number");
    }
    if(a < 0)
    {
        printf("Negative Number");
    }
}
```

Output

```
Enter Number:5
Positive Number
```

Output

```
Enter Number:-5
Negative Number
```

Modulus Operator

- `%` is modulus operator in C
- It divides the value of one expression (number) by the value of another expression (number), and returns the remainder.
- Syntax: `express1 % express2`
- E.g.
  - `7%2` Answer: 1
  - `6%2` Answer: 0
  - `25%10` Answer: 5
  - `37%28` Answer: 9

WAP to print Odd or Even Number

Program

```c
#include<stdio.h>
void main()
{
    int a;
    printf("Enter Number:");
    scanf("%d",&a);
    if(a%2 == 0)
    {
        printf("Even Number");
    }
    if(a%2 != 0)
    {
        printf("Odd Number");
    }
}
```

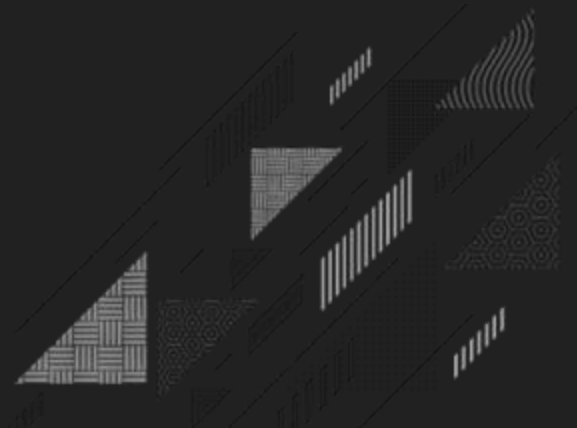
Output

```
Enter Number:12
Even Number
```

Output

```
Enter Number:11
Odd Number
```

If..else statement



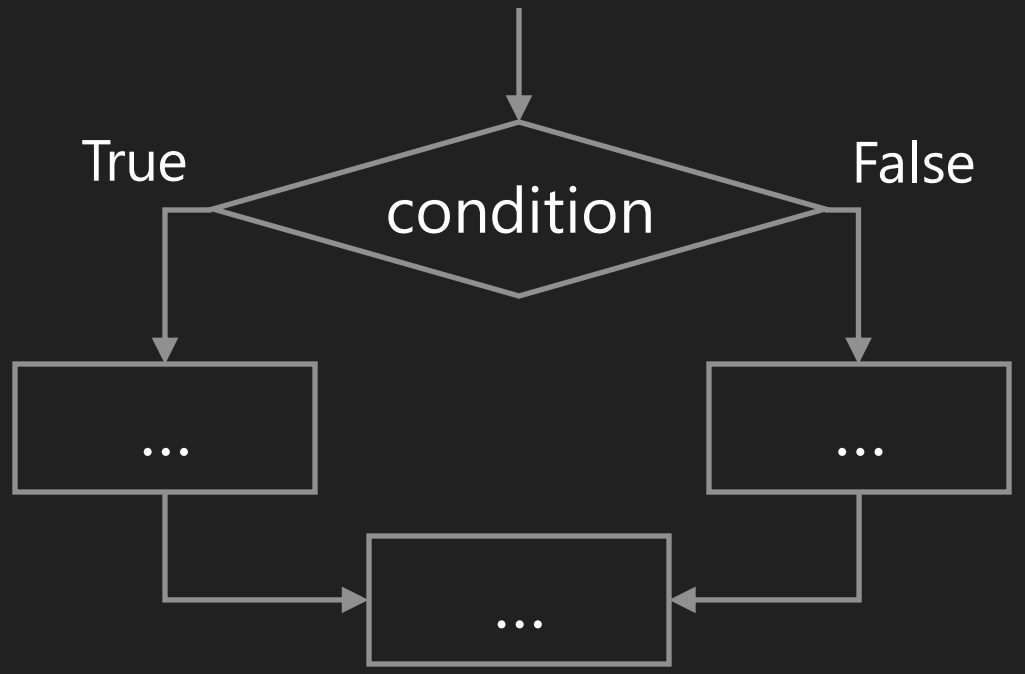
if...else

- **if…else** is two branch decision making statement
- If condition is true then true part will be executed else false part will be executed
- **else** is keyword

Syntax

```
if(condition)
{
    // true part
}
else
{
    // false part
}
```

Flowchart of **if…else**



WAP to print Positive or Negative Number using if...else

Program

```c
#include<stdio.h>
void main()
{
    int a;
    printf("Enter Number:");
    scanf("%d",&a);
    if(a >= 0)
    {
        printf("Positive Number");
    }
    else
    {
        printf("Negative Number");
    }
}
```

Output

```
Enter Number:5
Positive Number
```

Output

```
Enter Number:-5
Negative Number
```

WAP to print Odd or Even Number using if...else

Program

```c
#include<stdio.h>
void main()
{
    int a;
    printf("Enter Number:");
    scanf("%d",&a);
    if(a%2 == 0)
    {
        printf("Even Number");
    }
    else
    {
        printf("Odd Number");
    }
}
```

Output

```
Enter Number:12
Even Number
```

Output

```
Enter Number:11
Odd Number
```

WAP to find largest number from given 2 numbers using if

Program

```c
#include<stdio.h>
void main()
{
    int a, b;
    printf("Enter Two Numbers:");
    scanf("%d%d",&a,&b);
    if(a > b)
    {
        printf("%d is largest", a);
    }
    if(a < b)
    {
         printf("%d is largest", b);
    }
}
```

Output

```
Enter Two Numbers:4
5
5 is largest
```

WAP to find largest number from given 2 numbers using if...else

Program

```c
#include<stdio.h>
void main()
{
    int a, b;
    printf("Enter Two Numbers:");
    scanf("%d%d",&a,&b);
    if(a > b)
    {
        printf("%d is largest", a);
    }
    else
    {
         printf("%d is largest", b);
    }
}
```

Output

```
Enter Two Numbers:4
5
5 is largest
```

{ }

- If body of `if` contains only one statement then `{ }` are not compulsory
- But if body of `if` contains more than one statements then `{ }` are compulsory

```c
if(a >= b)
{
    printf("%d is largest", a);
}
```

Both are same

```c
if(a >= b)
    printf("%d is largest", a);
```

If...else if...else if...else Ladder if

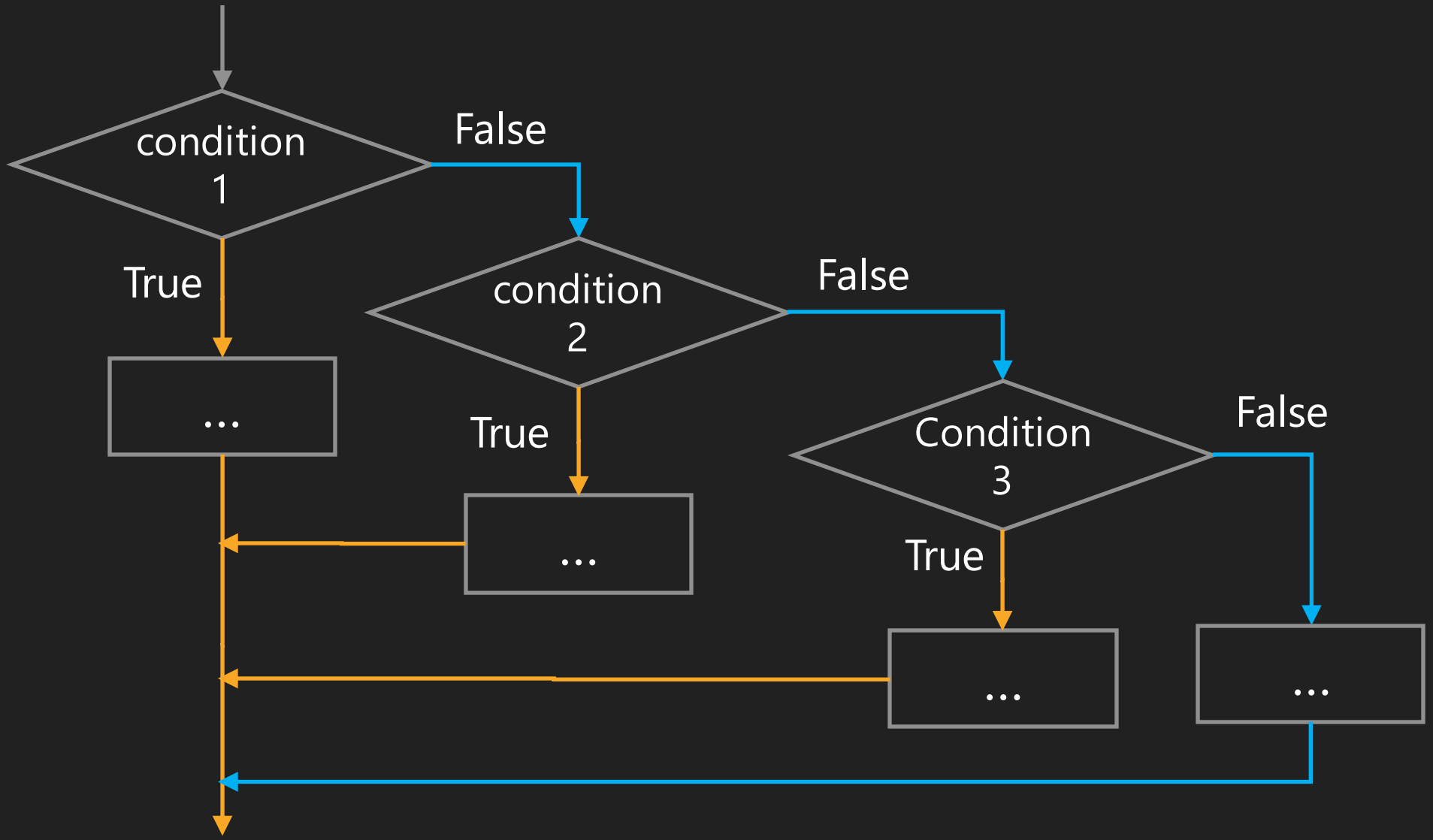
If…else if…else if…else

- **if…else if…else if…else** is multi branch decision making statement.
- If first **if** condition is true then remaining **if** conditions will not be evaluated.
- If first **if** condition is false then second **if** condition will be evaluated and if it is true then remaining **if** conditions will not be evaluated.
- **if…else if…else if…else** is also known as if…else if ladder

Syntax

```
if(condition-1)
  statement-1;
else if(condition-2)
  statement-2;
else
  statement-3;
```

`if…else` `if…else` ladder flowchart



WAP to print Zero, Positive or Negative Number

Program

```c
#include<stdio.h>
void main()
{
    int a;
    printf("Enter Number:");
    scanf("%d",&a);
    if(a > 0)
        printf("Positive Number");
    else if(a==0)
        printf("Zero");
    else
        printf("Negative Number");
}
```

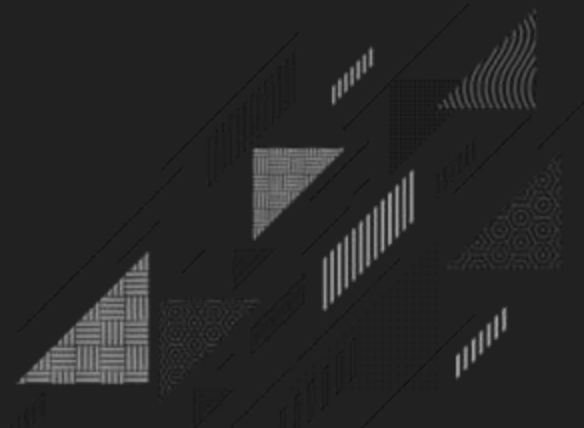
Output

```
Enter Number:5
Positive Number
```

Output

```
Enter Number:-5
Negative Number
```

Nested if



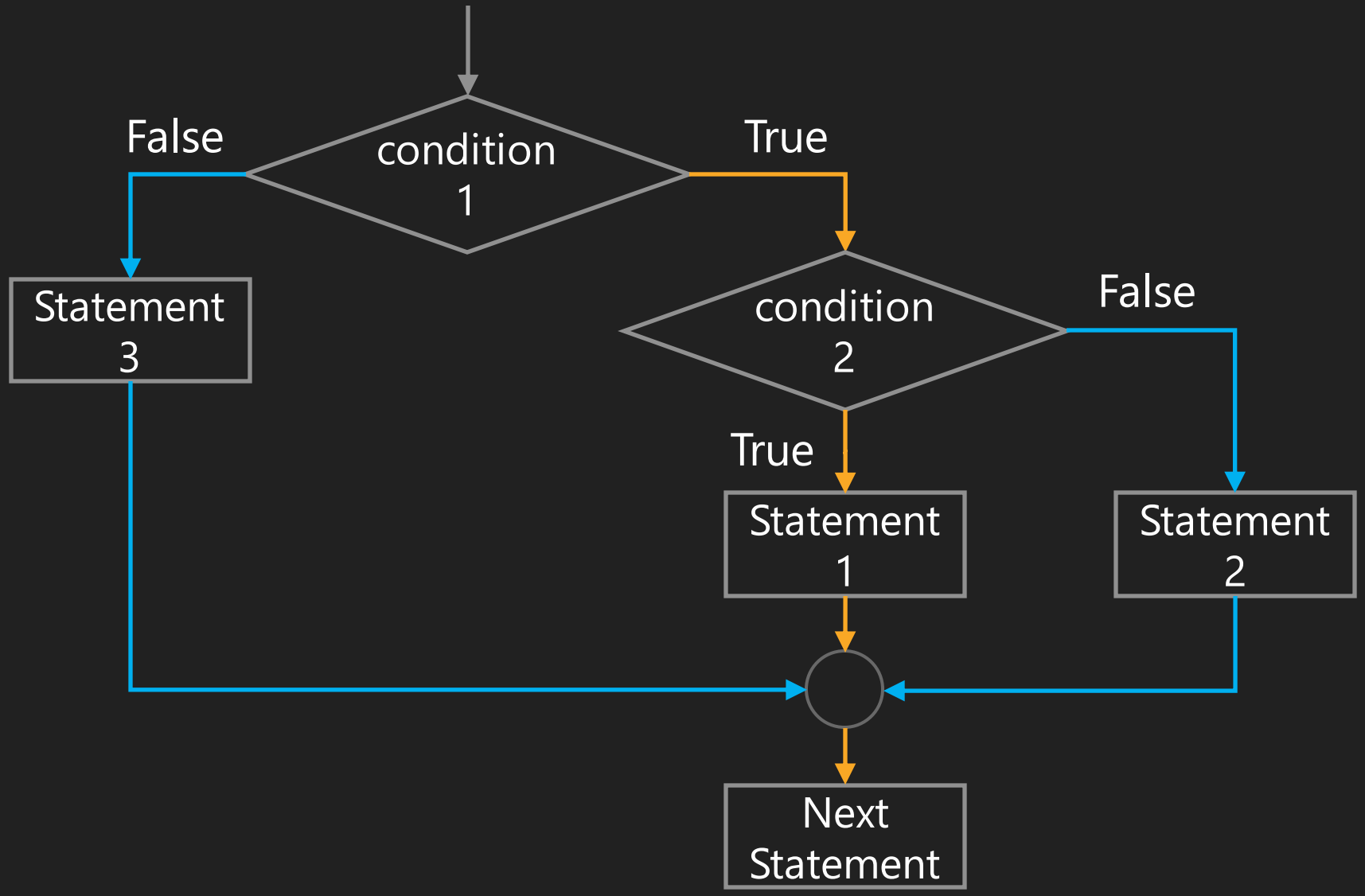
Nested if

- If condition-1 is true then condition-2 is evaluated. If it is true then statement-1 will be executed.
- If condition-1 is false then statement-3 will be executed.

Syntax

```
if(condition-1)
{
    if(condition-2)
    {
      statement-1;
    }
    else
    {
      statement-2;
    }
}
else
{
  statement-3;
}
```

Nested `if` flowchart



WAP to print maximum from given three numbers

Program

```c
void main(){
    int a, b, c;
    printf("Enter Three Numbers:");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b)
    {
      if(a>c)
        printf("%d is max",a);
      else
        printf("%d is max",c);
   }
   else
   {
     if(b>c)
       printf("%d is max",b);
     else
       printf("%d is max",c);
   }
}
```

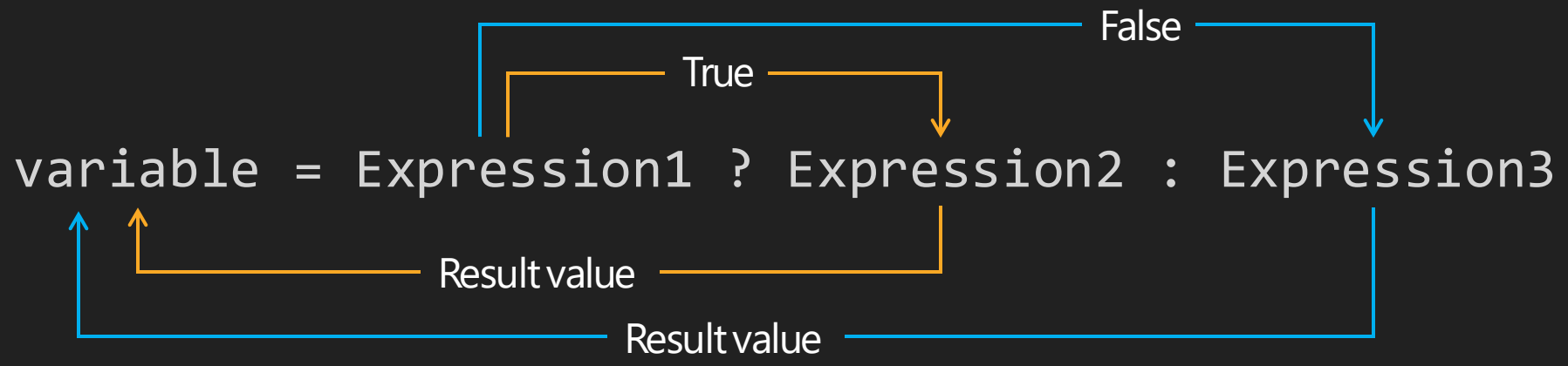
Output

```
Enter Three Numbers:7
5
9
9 is max
```

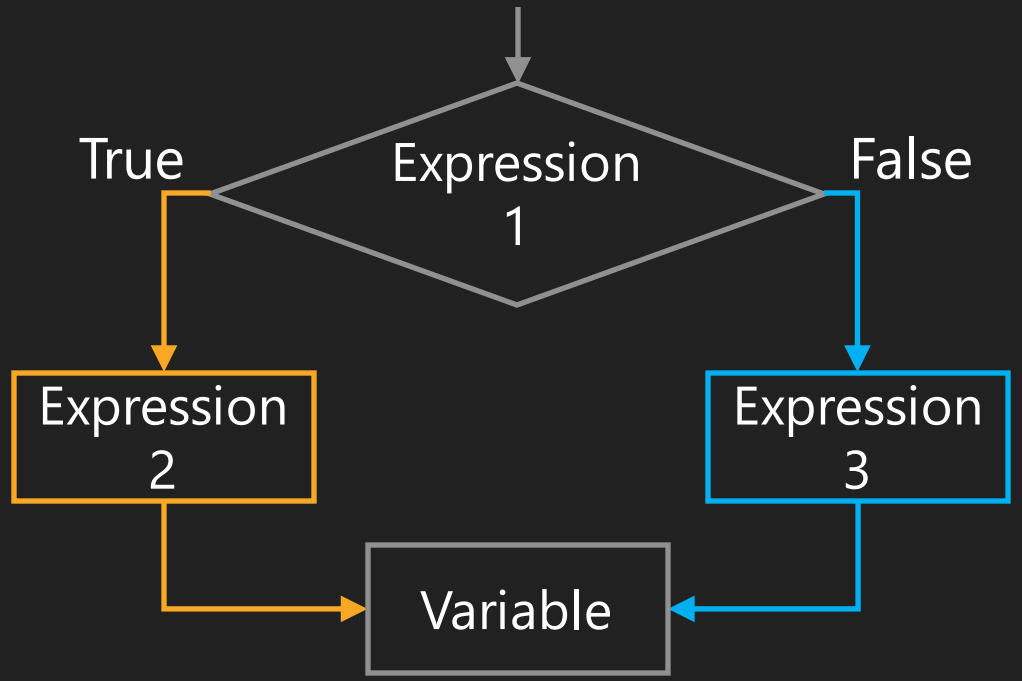
Conditional Operator

? : (Conditional Operator)

- The conditional works operator is similar to the if-else.
- It is also known as a ternary operator.
- It returns first value of expression (before colon(:)) if expression is true and second value of expression if expression is false.



Conditional operator flowchart



- Here, Expression1 is the condition to be evaluated.
- If the condition(Expression1) is True then Expression2 will be executed and the result will be returned.
- Otherwise, if condition(Expression1) is false then Expression3 will be executed and the result will be returned.

WAP to find largest number from given 2 numbers using ? :

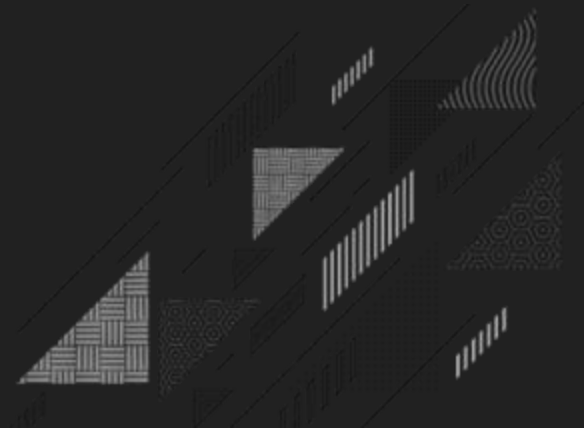
Program

```c
#include<stdio.h>
void main()
{
    int a, b, max;
    printf("Enter Two Numbers:");
    scanf("%d%d",&a,&b);
    max = a>b?a:b;
    printf("%d is largest",max);
}
```

Output

```
Enter Two Numbers:4
5
5 is largest
```

switch...case



switch...case

- The switch statement allows to execute one code block among many alternatives.
- It works similar to if...else..if ladder.

Syntax

```
switch (expression)
{
    case constant1:
      // statements
      break;
    case constant2:
      // statements
      break;
    .
    .
    .
    default:
      // default statements
}
```

- The expression is evaluated once and compared with the values of each `case`.
- If there is a match, the corresponding statements after the matching `case` are executed.
- If there is no match, the `default` statements are executed.
- If we do not use `break`, all statements after the matching label are executed.
- The `default` clause inside the `switch` statement is optional.

WAP that asks day number and prints day name using switch…case

```c
void main(){
    int day;
    printf("Enter day number(1-7):");
    scanf("%d",&day);
    switch(day)
    {
        case 1:
                printf("Sunday");
                break;
        case 2:
                printf("Monday");
                break;
        case 3:
                printf("Tuesday");
                break;
        case 4:
                printf("Wednesday");
                break;
        case 5:
                printf("Thursday");
                break;
        case 6:
                printf("Friday");
                break;
        case 7:
                printf("Saturday");
                break;
        default:
                printf("Wrong input");
                break;
    }
}
```

Output

```
Enter day number(1-7):5
Thursday
```

Practice programs

1) Write a program to check whether entered character is vowel or not?
2) Write a program to perform Addition, Subtraction, Multiplication and Division of 2 numbers as per user's choice (using if…else/Nested if/Ladder if).
3) Write a program to read marks of five subjects. Calculate percentage and print class accordingly. Fail below 35, Pass Class between 35 to 45, Second Class between 45 to 60, First Class between 60 to 70, Distinction if more than 70.
4) Write a program to find out largest number from given 3 numbers (Conditional operator).
5) Write a program to print number of days in the given month.

Thank you