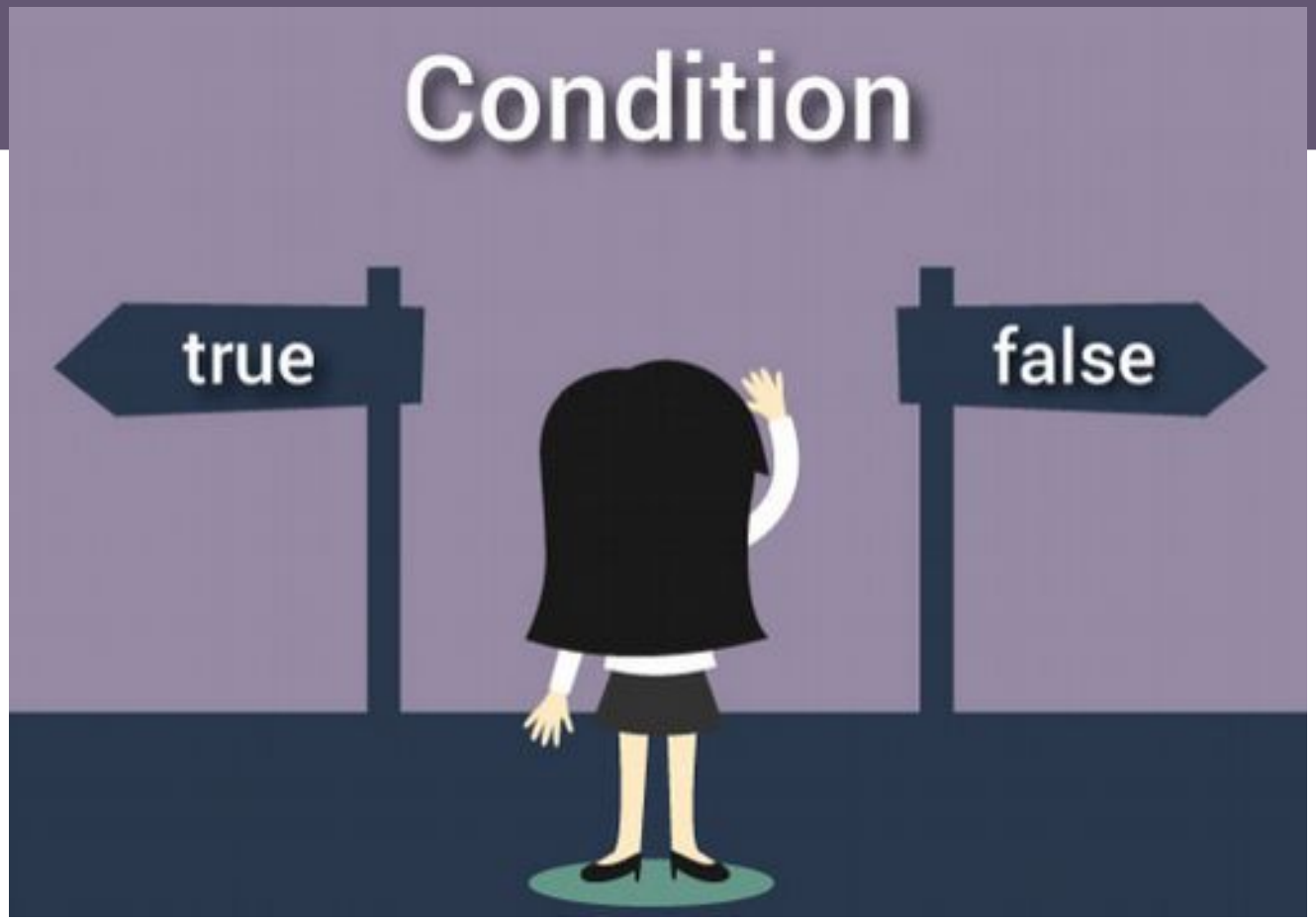
Problem Solving and Programming

Conditions

Kasun de Zoysa
kasun@ucsc.cmb.ac.lk

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING



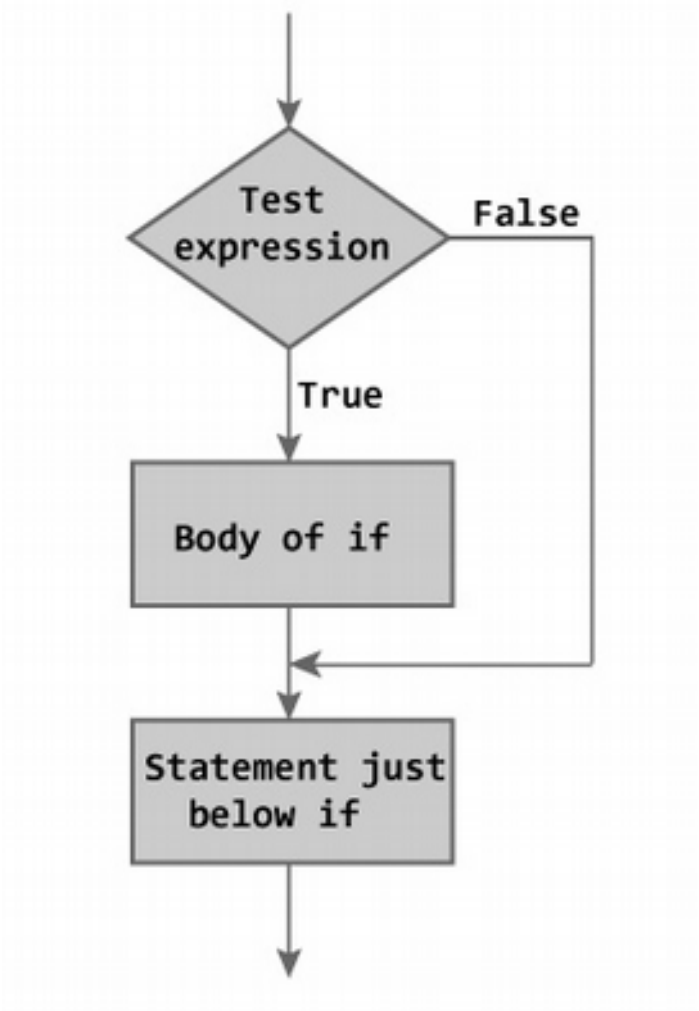
In programming, decision making is used to specify the order in which statements are executed.

Relational Operators

A relational operator checks the relationship between two operands. If the relation is true, it returns 1; if the relation is false, it returns value 0. Relational operators are used in decision making and loops.

Operator	Meaning of Operator	Example
==	Equal to	5 == 3 returns 0
>	Greater than	5 > 3 returns 1
<	Less than	5 < 3 returns 0
!=	Not equal to	5 != 3 returns 1
>=	Greater than or equal to	5 >= 3 returns 1
<=	Less than or equal to	5 <= 3 return 0

C if statement




```
if (testExpression)
{
    // statements
}
```

The if statement evaluates the test expression inside the parenthesis.

If the test expression is evaluated to true (nonzero), statements inside the body of if is executed.

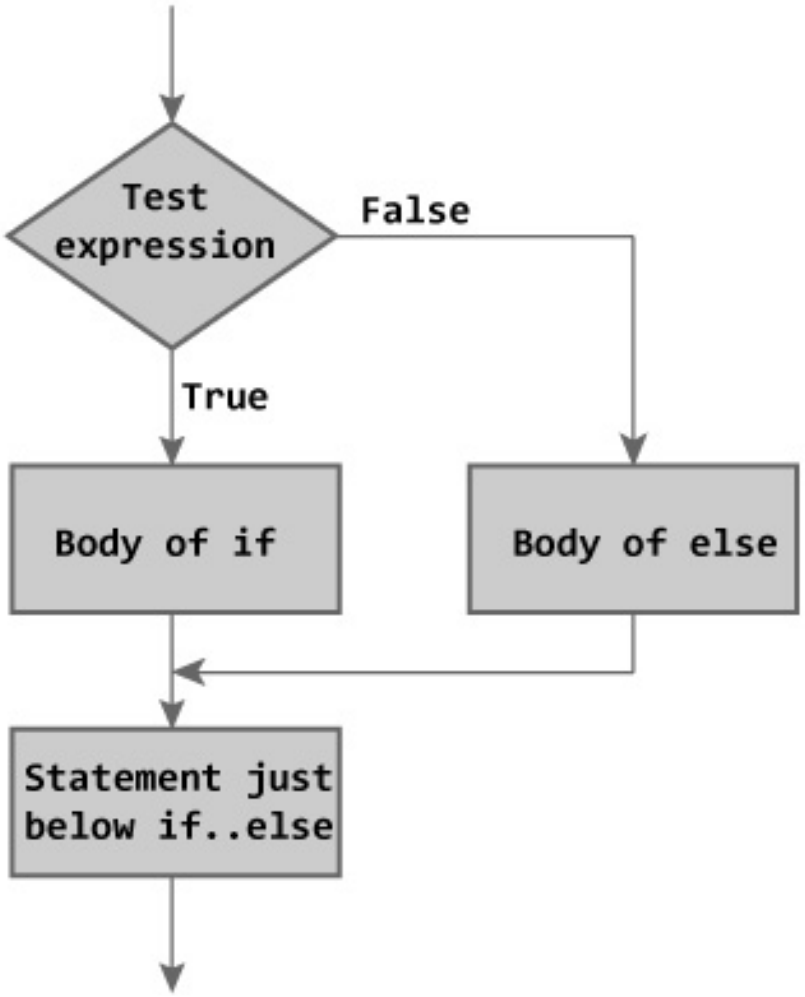
If the test expression is evaluated to false (0), statements inside the body of if is skipped from execution.

C if...else statement

```
if (testExpression) {
    // codes inside the body of if
}
else {
    // codes inside the body of else
}
```

If test expression is true, codes inside the body of if statement is executed and, codes inside the body of else statement is skipped.

If test expression is false, codes inside the body of else statement is executed and, codes inside the body of if statement is skipped.



Nested if...else statement (if...else if....else Statement)

```
if (testExpression1)
{
    // statements to be executed
}
else if(testExpression2)
{
    // statements to be executed
}
else if (testExpression 3)
{
    // statements to be executed
}
.
.
else
{
    // statements to be executed
}
```

Logical Operators

An expression containing logical operator returns either 0 or 1 depending upon whether expression results true or false. Logical operators are commonly used in decision making in C programming.

Operator	Meaning of Operator	Example
&&	Logial AND. True only if all operands are true	If c = 5 and d = 2 then, expression `((c == 5) && (d > 5))` equals to 0.
||	Logical OR. True only if either one operand is true	If c = 5 and d = 2 then, expression `((c == 5) || (d > 5))` equals to 1.
!	Logical NOT. True only if the operand is 0	If c = 5 then, expression `! (c == 5)` equals to 0.

```c
// C Program to demonstrate the working of logical operators

#include <stdio.h>
int main()
{
    int a = 5, b = 5, c = 10, result;

    result = (a == b) && (c > b);
    printf("(a == b) && (c > b) equals to %d \n", result);

    result = (a == b) && (c < b);
    printf("(a == b) && (c < b) equals to %d \n", result);

    result = (a == b) || (c < b);
    printf("(a == b) || (c < b) equals to %d \n", result);

    result = (a != b) || (c < b);
    printf("(a != b) || (c < b) equals to %d \n", result);

    result = !(a != b);
    printf("!(a == b) equals to %d \n", result);

    result = !(a == b);
    printf("!(a == b) equals to %d \n", result);

    return 0;
}
```

Output

```
(a == b) && (c > b) equals to 1
(a == b) && (c < b) equals to 0
(a == b) || (c < b) equals to 1
(a != b) || (c < b) equals to 0
!(a != b) equals to 1
!(a == b) equals to 0
```

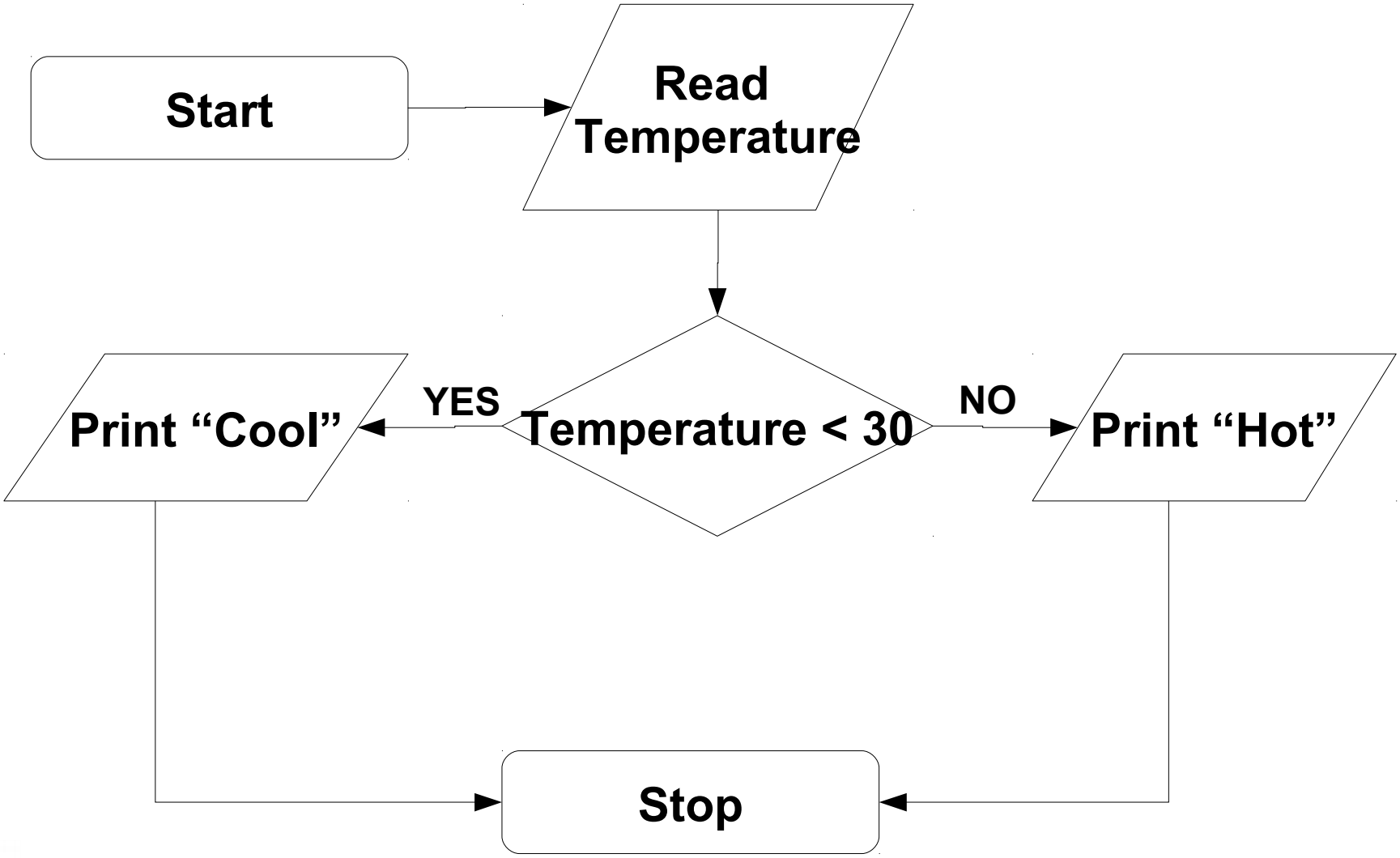
C Ternary Operator (?:)

The conditional operator works as follows:
1. The first expression conditionalExpression is evaluated first. This expression evaluates to 1 if it's true and evaluates to 0 if it's false.
2. If conditionalExpression is true, expression1 is evaluated.
3. If conditionalExpression is false, expression2 is evaluated.

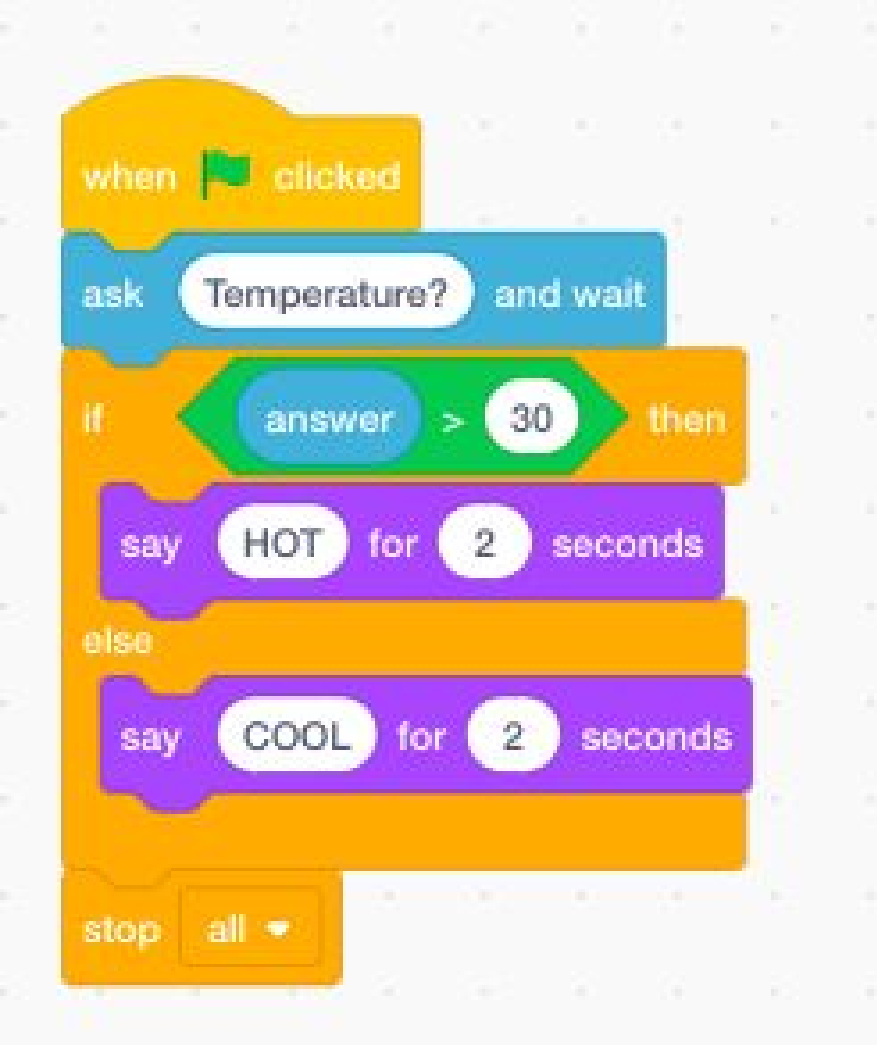
```c
#include <stdio.h>
int main(){
    char February;
    int days;
    printf("If this year is leap year, enter 1. If not enter any integer: ")
    scanf("%c",&February);

    // If test condition (February == 'l') is true, days equal to 29.
    // If test condition (February =='l') is false, days equal to 28.
    days = (February == '1') ? 29 : 28;

    printf("Number of days in February = %d",days);
    return 0;
}
```

Problem-4: Write a program to read temperature and print “Cool” or “Hot”

If temperature is less than 30: Cool Otherwise: Hot



Program-4



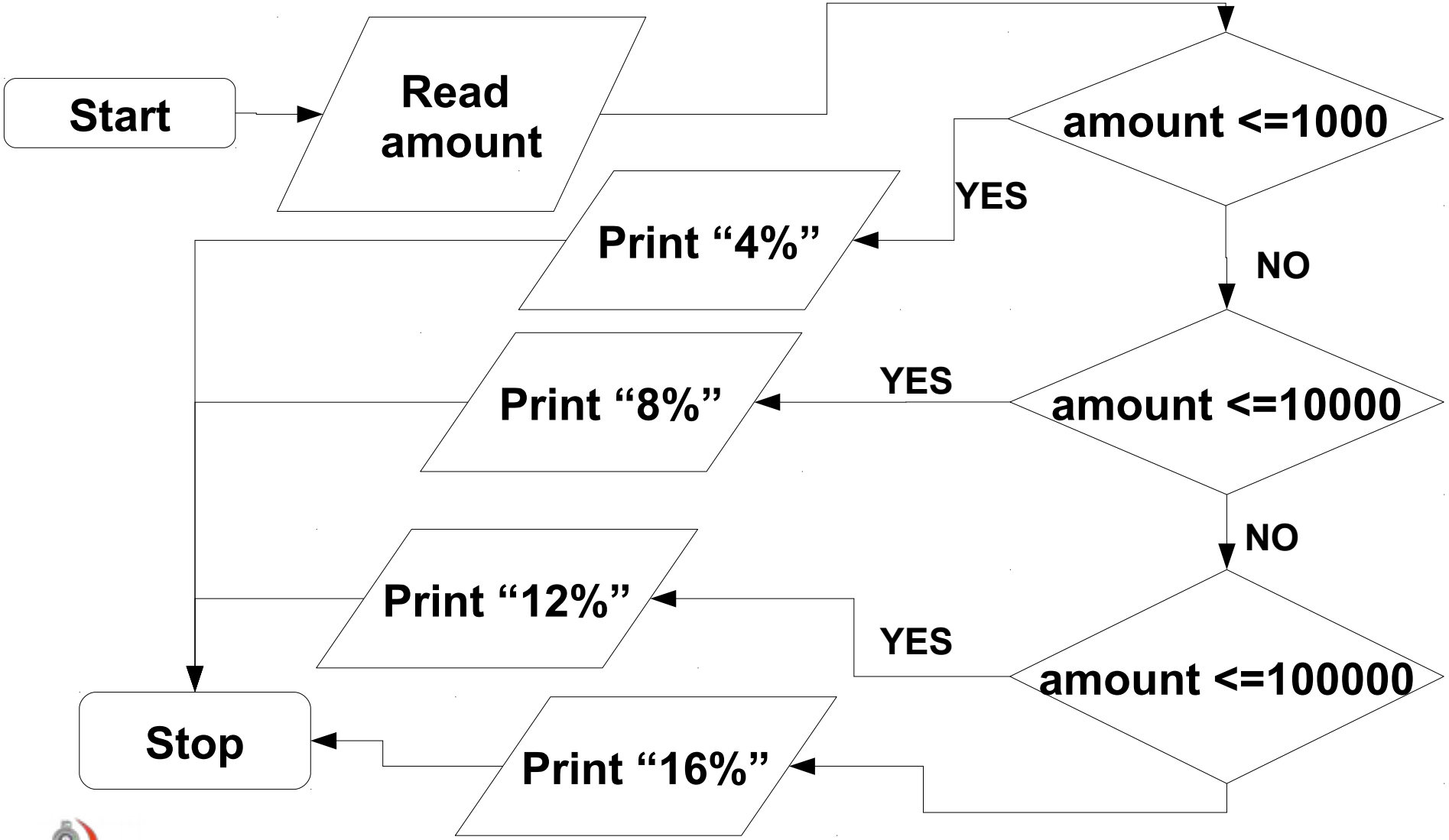

```
#include <stdio.h>

int main(){
      int tmp=0;

      printf("Enter tmp >");
      scanf("%d",&tmp);
      if(tmp>=30) printf("HOT\n");
       else printf("COOL\n");

 return 0;
}
```

Bank Interest

Write a program show the interest rate when deposit amount is given.

The bank pays the interest as follows:
a flat 4% for deposits of up to Rs. 1000,
a flat 8% per year for deposits of up to Rs. 10000,
a flat 12% per year for deposits of up to Rs. 100000,
and a flat 16% for deposits of more than Rs. 100000.

Problem: Bank Interest



Problems

1) Write a C program that takes a number from the user and checks whether that number is either positive or negative or zero.

2) Write a C program that takes a number from the user and checks whether that number is odd or even.

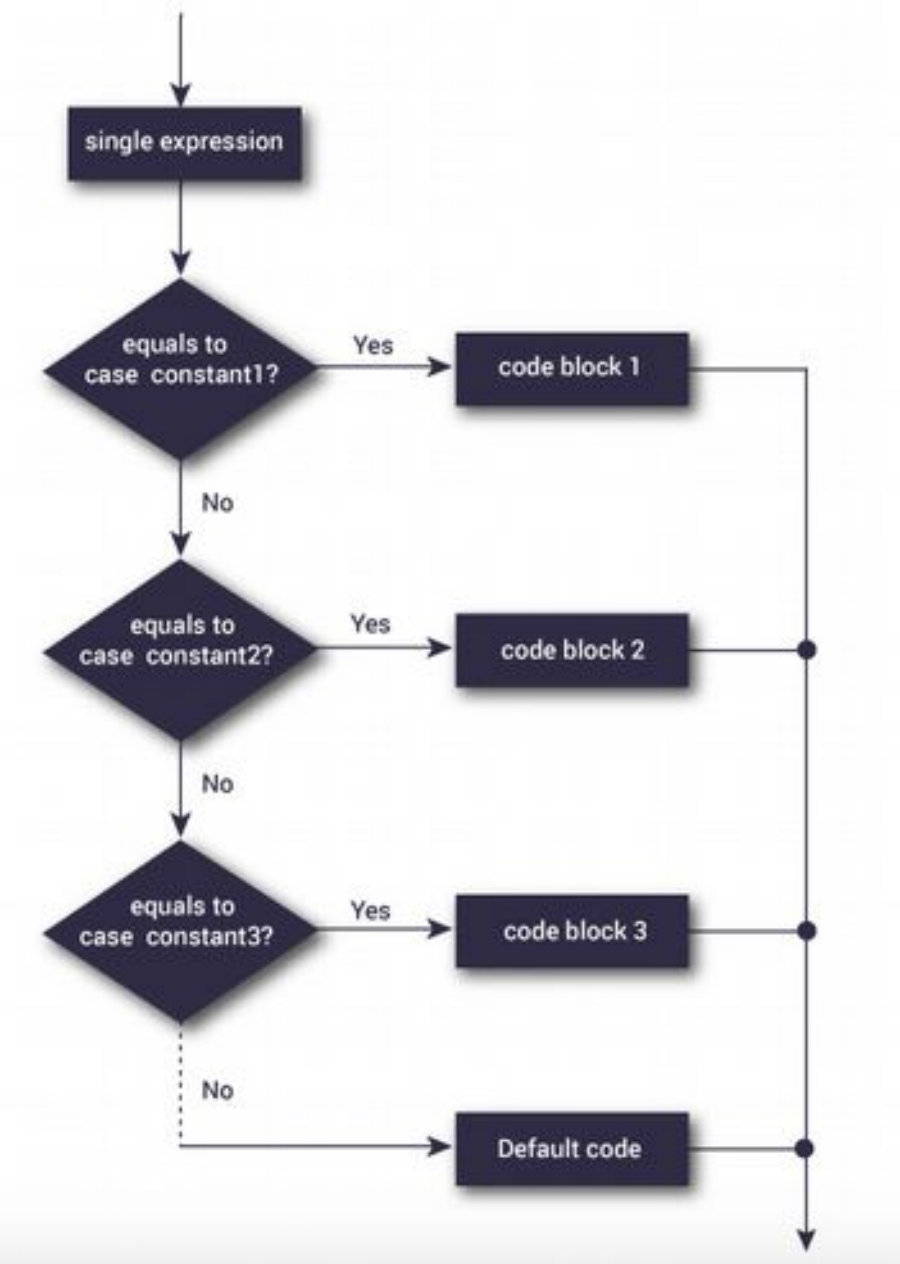
3) Write a C program to check a given character is Vowel or Consonant.

Odd and Even Numbers

```c
// Program to check whether an integer entered by the user is odd or even

#include <stdio.h>
int main()
{
      int number;
      printf("Enter an integer: ");
      scanf("%d",&number);

      // True if remainder is 0
      if( number%2 == 0 )
            printf("%d is an even integer.",number);
      else
            printf("%d is an odd integer.",number);
      return 0;
}
```

C switch...case Statement

The if..else..if ladder allows you to execute a block code among many alternatives.
If you are checking on the value of a single variable in if...else...if, it is better to use switch statement.
The switch statement is often faster than nested if...else.


single expression
equals to case constant1?
Yes
code block 1
No
equals to case constant2?
Yes
code block 2
No
equals to case constant3?
Yes
code block 3
No
Default code

Syntax of switch...case

```
switch (n)
{
     case constant1:
          // code to be executed if n is equal to constant1;
          break;

     case constant2:
          // code to be executed if n is equal to constant2;
          break;
          .
          .
          .
     default:
          // code to be executed if n doesn't match any constant
}
```

Discussion