ESC101: Introduction to Computing

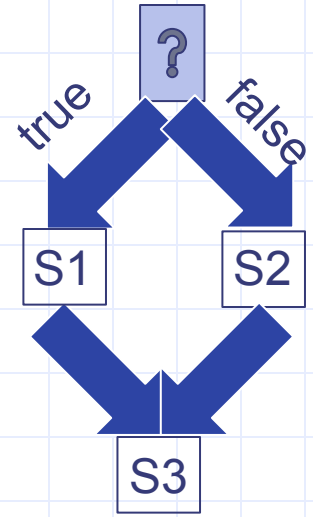
Conditional Statements

Conditional statements in C

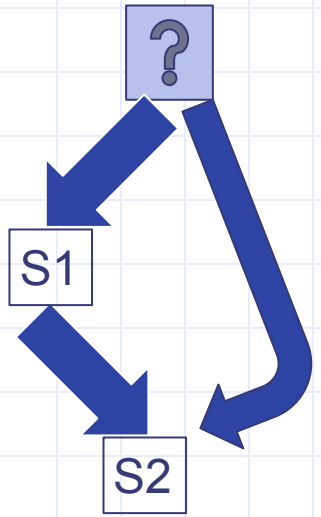
- Three types of conditional statements in C
  - if (condition) *action*
    else *some-other-action*
  - if (condition) *action*
  - switch-case
- Each action is a sequence of one or more statements

if statement

```
if (expression)
        statement S1
else
        statement S2
statement S3
```




```
if (expression)
        statement S1
statement S2
```



Nested if, if-else

```
if (a <= b) {
        if (a <= c) { … }  else {…}
} else {
         if (b <= c)  { … } else { … }
}
```

- if and if-else are also statements, they can be used anywhere a statement or block can be used.

Else if

- A special kind of nesting is the chain of if-else-if-else-… statements

```
if (cond1) {
     stmt1
} else {
       if (cond2)  {
            stmt2
       } else {
           if (cond3) {
               ….
           }
       }
}
```

General form of if-else-if-else…

```
if (cond1)
     {stmt-block1}
else if (cond2)
     {stmt-block2}
else if (cond3)
     {stmt-block3}
else if (cond4)
     {stmt-block4}
else if  …
else
      last-block-of-stmt
```

if-else

```
if ((a != 0) && (b != 0))
    if (a * b >= 0)
        printf (“positive”);
else
    printf(“zero”);
```

OUTPUT for a = 5, b = 0

NO OUTPUT!!

OUTPUT for a = 5, b = -5

zero

```
if ((a != 0) && (b != 0))
    if (a * b >= 0)
        printf (“positive”);
    else
        printf(“negative”);
```

OUTPUT for a = 5, b = 0

NO OUTPUT!!

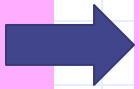
OUTPUT for a = 5, b = -5

negative

Unmatched if and else

- An else always matches closest unmatched if
  - Unless forced otherwise using { … }

```
if (cond1)
    if (cond2)
        …
   else
        …
```



```
if (cond1) {
    if (cond2)
        …
    else
        …
}
```

Unmatched if and else

- An else always matches closest unmatched if
  - Unless forced otherwise using { … }

```
if (cond1)
   if (cond2)
      …
   else
      …
```

IS NOT SAME AS

```
if (cond1) {
    if (cond2)
        …
}
else
        …
```

Write a program that checks if product of 2 numbers is positive, negative or 0

```
 # include <stdio.h>
int main(){
   float a, b;
   scanf("%f %f",&a, &b);
   if( _________________ )
      if( __________ )
         printf("positive\n");
      else
         printf("negative\n");
   else
      printf("zero\n");
   return 0;
}
```

Write a program that checks if product of 2 numbers is positive, negative or 0

```
 # include <stdio.h>
int main(){
   float a, b;
   scanf("%f %f",&a, &b);
   if( ________________ )
      if( a*b >=0 )
         printf("positive\n");
      else
         printf("negative\n");
   else
      printf("zero\n");
   return 0;
}
```

Write a program that checks if product of 2 numbers is positive, negative or 0

```
 # include <stdio.h>
int main(){
   float a, b;
   scanf("%f %f",&a, &b);
   if(  a != 0.0 && b != 0.0 )
      if( a*b >=0 )
         printf("positive\n");
      else
         printf("negative\n");
   else
      printf("zero\n");
   return 0;
}
```

Write a program that checks if product of 2 numbers is positive, negative or 0

```
# include <stdio.h>
int main(){
  float a, b;
  scanf("%f %f",&a, &b);
  if( _______________ )
      printf( "zero\n");
 else if (_____________)
      printf( "positive\n");
 else
      printf( "negative\n");
   return 0;
}
```

Write a program that checks if product of 2 numbers is positive, negative or 0

```
# include <stdio.h>
int main(){
  float a, b;
  scanf("%f %f",&a, &b);
  if(  a == 0.0 ||  b == 0.0 )
      printf( "zero\n");
 else if (a*b > 0.0)
      printf( "positive\n");
 else
      printf( "negative\n");
   return 0;
}
```

A problem

- Write a program to check if a year is a leap year

- A leap year is exactly divisible by 4 except for century years (years ending with 00). The century year is a leap year only if it is perfectly divisible by 400.

Write a program that checks if a year is a leap year

```
 #include <stdio.h>
int main()
{
   int year;
   scanf("%d",&year);
   if(year % 4 == 0)
      printf("%d is a leap year\n",year);
   else if(year % 100 == 0)
      printf("%d is not a leap year\n",year);
   else if(year % 400 == 0)
      printf("%d is a leap year\n",year);
   else
     printf("%d is not a leap year\n",year);
   return 0;
}
```

Input: 2012 Output: 2012 is a leap year

Write a program that checks if a year is a leap year

```
 #include <stdio.h>
int main()
{
   int year;
   scanf("%d",&year);
   if(year % 4 == 0)
      printf("%d is a leap year\n",year);
   else if(year % 100 == 0)
      printf("%d is not a leap year\n",year);
   else if(year % 400 == 0)
      printf("%d is a leap year\n",year);
   else
     printf("%d is not a leap year\n",year);
   return 0;
}
```

Input: 1900 Output: 1900 is a leap year **INCORRECT**

Write a program that checks if a year is a leap year

```
 #include <stdio.h>
int main()
{
   int year;
   scanf("%d",&year);
   if(year % 4 == 0)
      if(year % 100 == 0)
         if(year % 400 == 0)
            printf("%d is a leap year\n",year);
         else
            printf("%d is not a leap year\n",year);
      else
         printf("%d is a leap year\n",year);
   else
      printf("%d is not a leap year\n",year);
   return 0;
}
```

Write a program that checks if a year is a leap year

```
 #include <stdio.h>
int main()
{
   int year;
   scanf("%d",&year);
   if(year % 400 == 0)
      printf("%d is a leap year\n",year);
   else if(year % 100 == 0)
      printf("%d is not a leap year\n",year);
   else if(year % 4 == 0)
      printf("%d is a leap year\n",year);
   else
     printf("%d is not a leap year\n",year);
   return 0;
}
```

Write a program that checks if a year is a leap year

```
#include <stdio.h>
int main()
{
   int year;
   scanf("%d",&year);
   if( (year % 400 == 0) || ( (year%4 == 0)
         &&  (year%100 !=0) ) )
      printf("%d is a leap year\n",year);
   else
     printf("%d is not a leap year\n",year);
   return 0;
}
```

Printing the day

```
int day;
scanf (“%d”, &day);
if (day == 1) { printf(“Sunday”); }
else if (day == 2) { printf (“Monday”); }
else if (day == 3) { printf (“Tuesday”); }
else if (day == 4) { printf (“Wednesday”); }
else if (day == 5) { printf (“Thursday”); }
else if (day == 6) { printf (“Friday”); }
else if (day == 7) { printf (“Saturday”); }
else { printf (“ Illegal day %d”, day); }
```

Switch-case statement

- Multi-way decision
- Checks whether an expression matches one out of a number of constant integer (or char) values
- Execution *branches* based on the match found

Printing the day, version 2

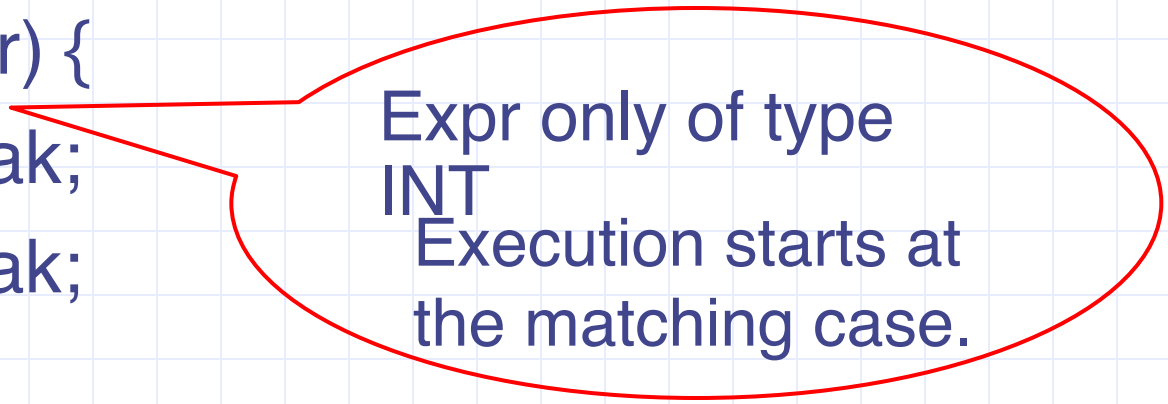
```
switch (day) {
case 1: printf(“Sunday”); break;
case 2: printf (“Monday”); break;
case 3: printf (“Tuesday”); break;
case 4: printf (“Wednesday”); break;
case 5: printf (“Thursday”); break;
case 6: printf (“Friday”); break;
case 7: printf (“Saturday”); break;
default: printf (“ Illegal day %d”, day);
}
```

Weekday, version 4

```
switch (day) {
case 1:
case 7: printf (“Weekend”); break;
case 2:
case 3:
case 4:
case 5:
case 6: printf (“Weekday”); break;
default: printf (“ Illegal day %d”, day);
}
```

General Form of switch-case

```
switch (selector-expr) {
case label1: s1; break;
case label2: s2; break;
...
case labelN: sN; break;
default : sD;
}
```



- **default** is optional. (= *remaining cases*)
- The location of **default** does not matter.
- The statements following a case label are executed one after other until a **break** is encountered (**Fall Through**)

Fall Through…

```
int n = 100;
int digit = n%10; // last digit
switch (digit) {
default : printf("Not divisible by 5\n");
          break;
case 0: printf("Even\n");
case 5: printf("Divisible by 5\n");
        break;
}
```

What is printed by the program fragment?

Answer:
Even
Divisible by 5

A problem

- Write a program to check if a number is even or odd using switch statement

Write a program that checks if a number is even or odd

```
 #include <stdio.h>
int main()
{
   int n;
   scanf("%d",&n);
   switch(____)
   {
      case 0:
         printf("%d is Even\n",n);
         break;
      case 1:
         printf("%d is Odd\n",n);
         break;
      default: printf("invalid number\n");
   }
   return 0;
}
```

Write a program that checks if a number is even or odd

```
 #include <stdio.h>
int main()
{
   int n;
   scanf("%d",&n);
   switch(n%2)
   {
      case 0:
         printf("%d is Even\n",n);
         break;
      case 1:
         printf("%d is Odd\n",n);
         break;
      default: printf("invalid number\n");
   }
   return 0;
}
```

Write a program that checks if a number is even or odd

```
 #include <stdio.h>
int main()
{
   int n;
   scanf("%d",&n);
   switch(n%2)
   {
      case 0:
         printf("%d is Even\n",n);
         break;
      default:
         printf("%d is Odd\n",n);
   }
   return 0;
}
```

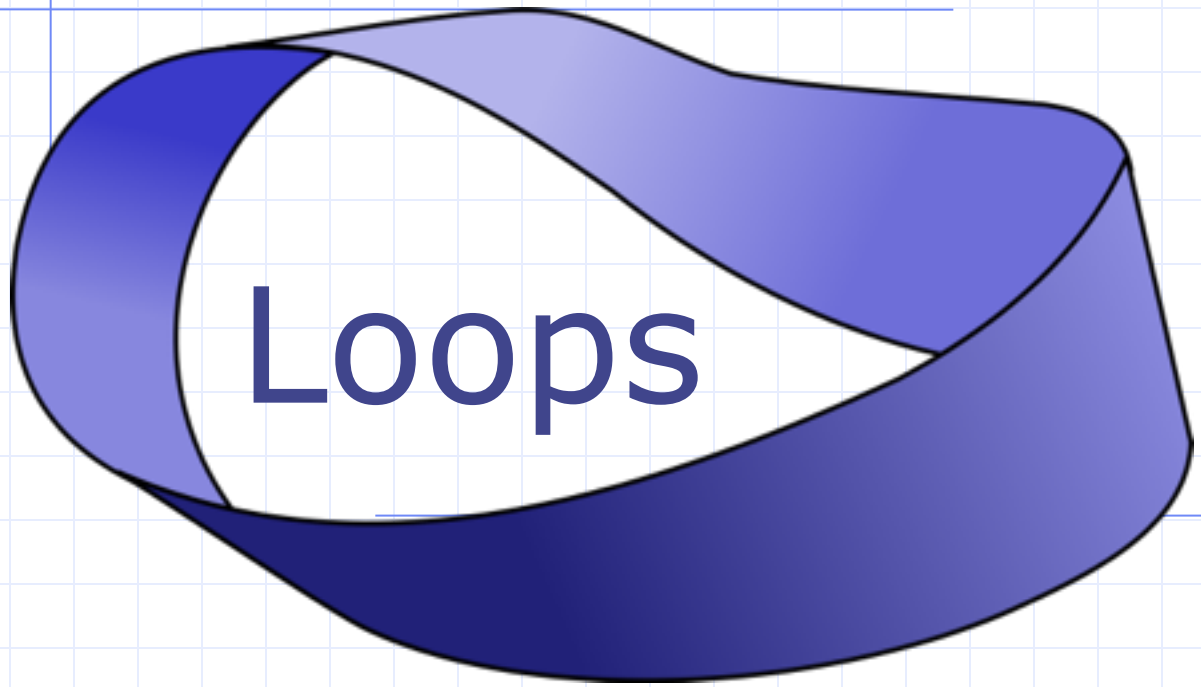
Write a program that checks if a character is capital or not using switch

```
#include <stdio.h>
int main()
{
        char c;
        scanf("%c",&c);
        switch(c)
        {
                case (c>= 'A' && c <='Z'):
                        printf("Capital\n");
                        break;
                case c>= 'a' && c<='z':
                        printf("small\n");
                        break;
                case c>='0' && c<='9':
                        printf("digit\n");
                        break;
                default: printf("illegal character\n");
                        break;
        }
        return 0;
}
```

WRONG

One can have only constant case and not logical expressions

ESC101: Introduction to Computing

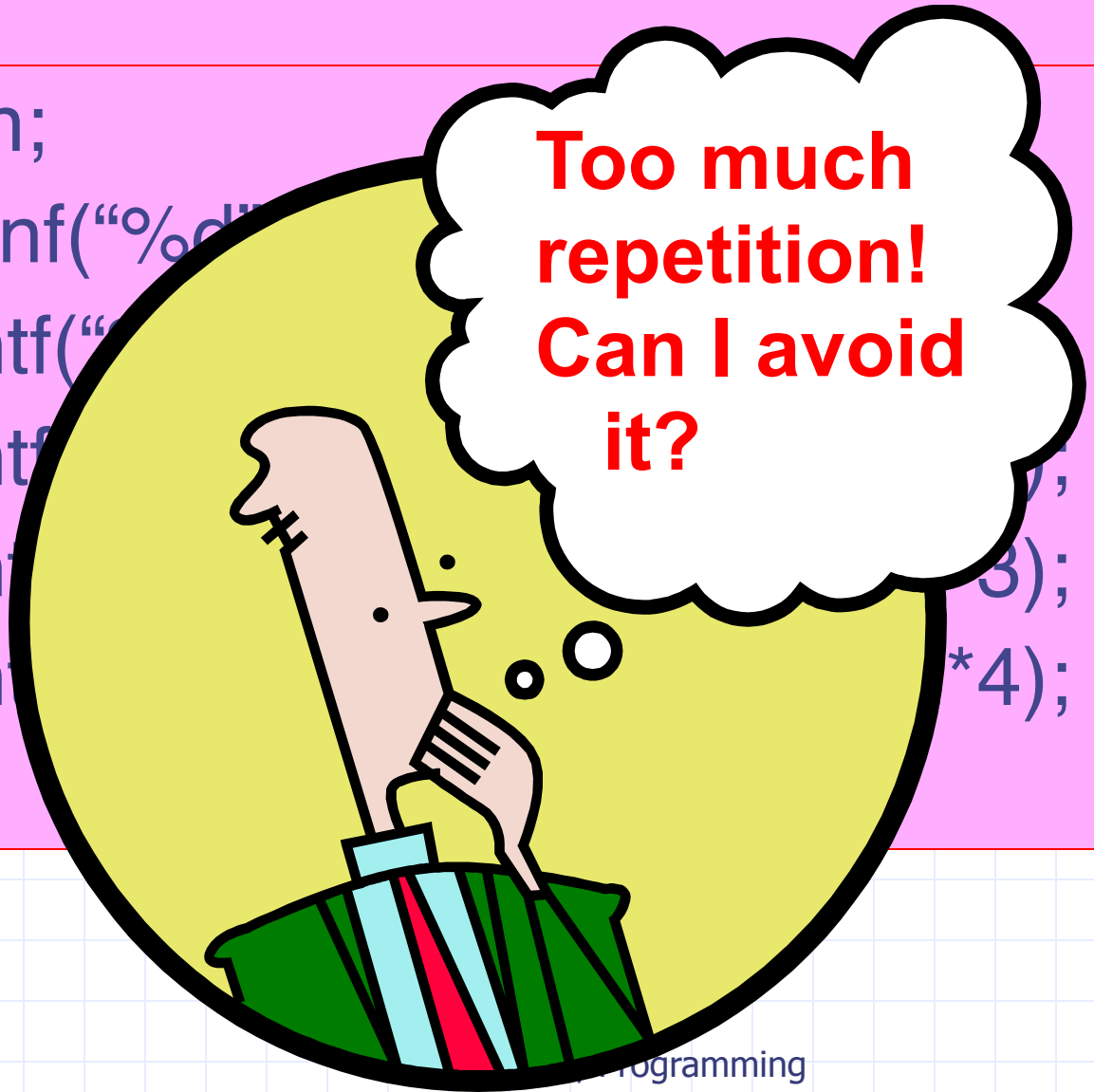


Printing Multiplication Table

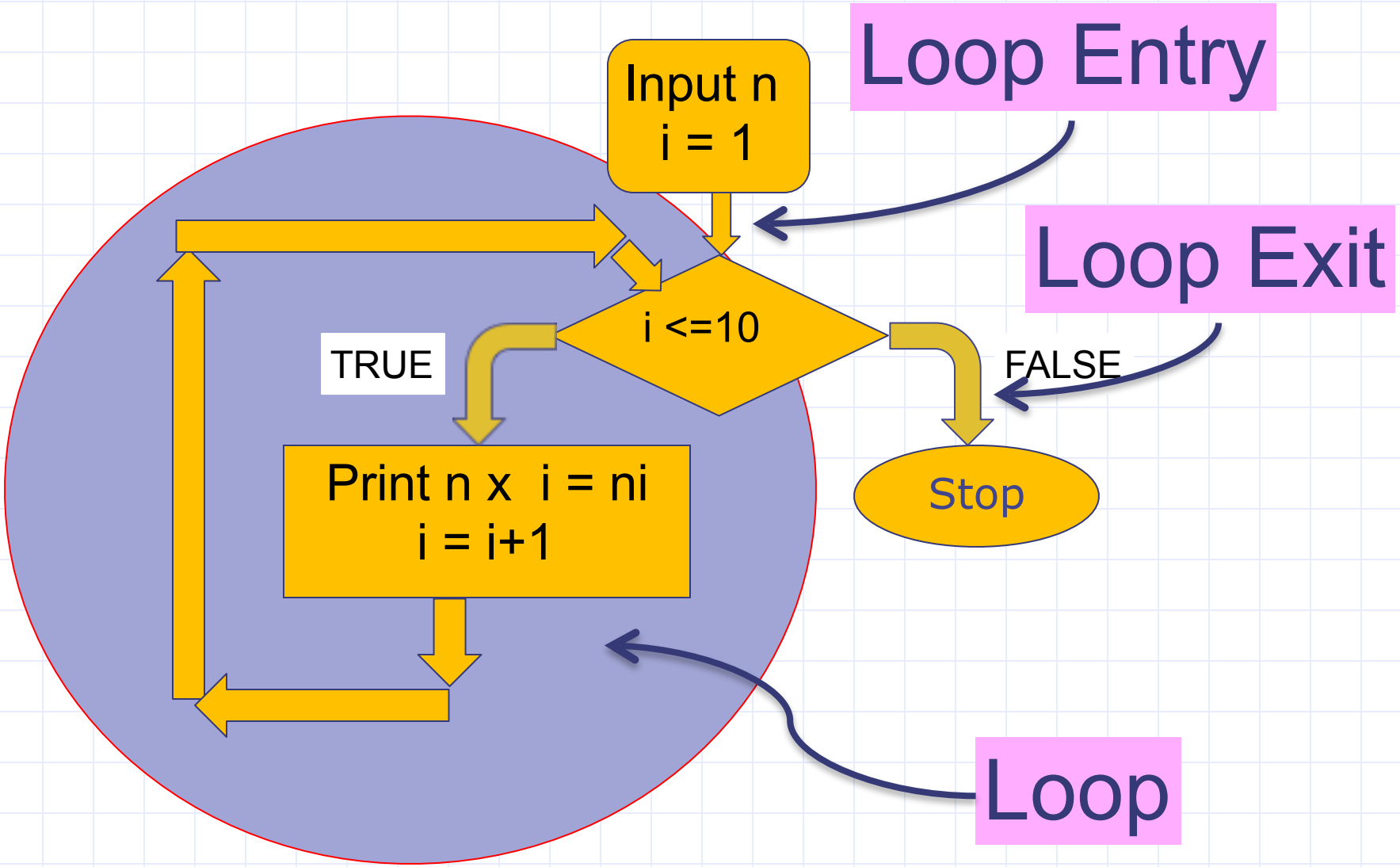
5	X	1	=	5
5	X	2	=	10
5	X	3	=	15
5	X	4	=	20
5	X	5	=	25
5	X	6	=	30
5	X	7	=	35
5	X	8	=	40
5	X	9	=	45
5	X	10	=	50

Program…

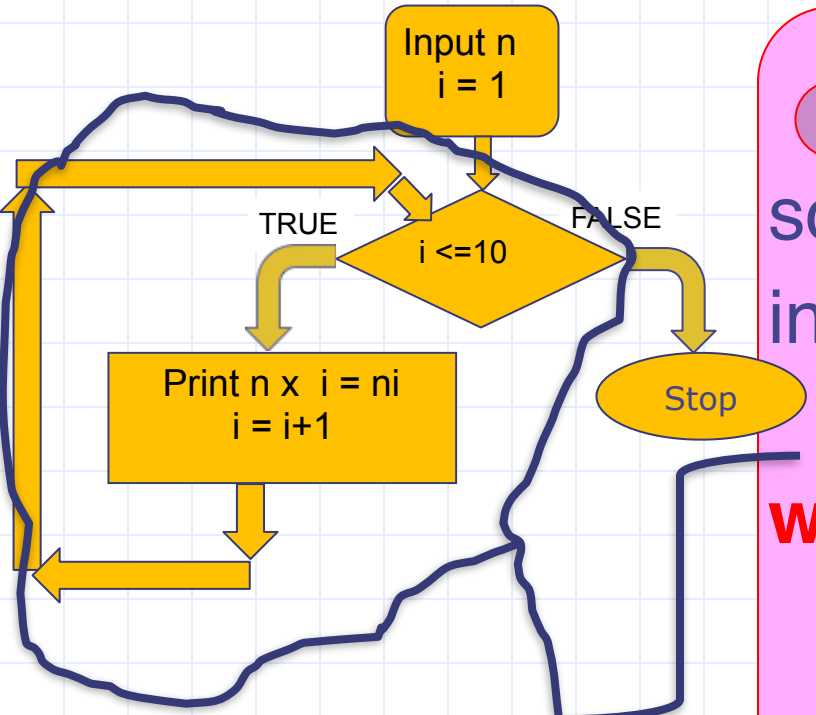
```
int n;
scanf("%d
printf("
printf
print
print
....
```



Printing Multiplication Table



Printing Multiplication Table



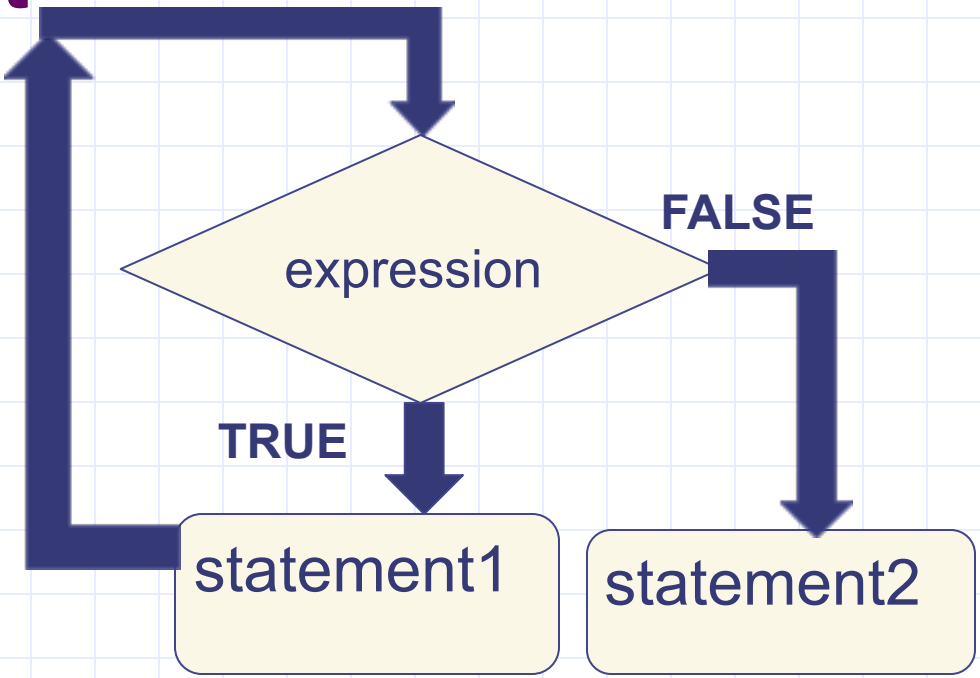

```
scanf("%d", &n);
int i = 1;

while (i <= 10) {
   printf("%d X %d = %d",
              n, i, n*i);
   i = i + 1;
}

// loop exited!
```

While Statement

```
while (expression)
    statement1;
statement2;
```



As long as expression is TRUE execute statement1.

When expression becomes FALSE execute statement 2.