Finish the first C program
Redirect input/output
File operation

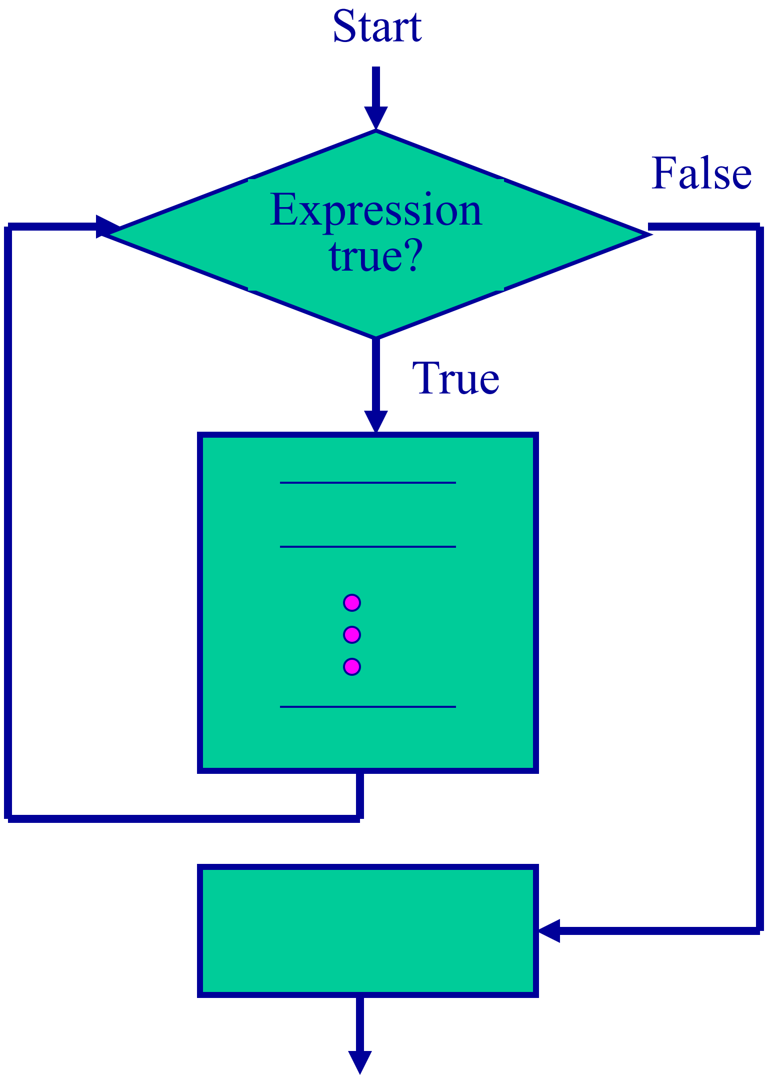
- The while Statement
- The do while Statement
- Example
- The if Statement
- More on the if Statement

The while Statement

The form of the while statement is

```
while ( expression )
{
        action
}
```

The right figure is the flowchart of the while statement



```
#include <stdio.h>
main ( )
{
      int x;
      x = 8;
      while ( x >= 0 )
      {
               printf ( “x = %d\n”, x );
               x = x - 3;
      }
}
```

The output is

```
           x = 8

           x = 5

           x = 2
```

```
#include <stdio.h>
main ( )
{
      int x;
      x = 0;
      while ( x <= 7 )
      {
            printf ( “x = %d\n”, x );
            x = x + 3;
      }
}
```

The output is

```
x = 0

x = 3

x = 6
```

```
#include <stdio.h>
main ( )
{
    int x;
    x = 0;
    while ( x <= 9 )
            x = x + 3;
    printf ( “total = %d\n”, x );
}
```

The output is

```
total = 12
```

No {} are needed if there is only one action statement

The do while Statement

The form of the do while statement is

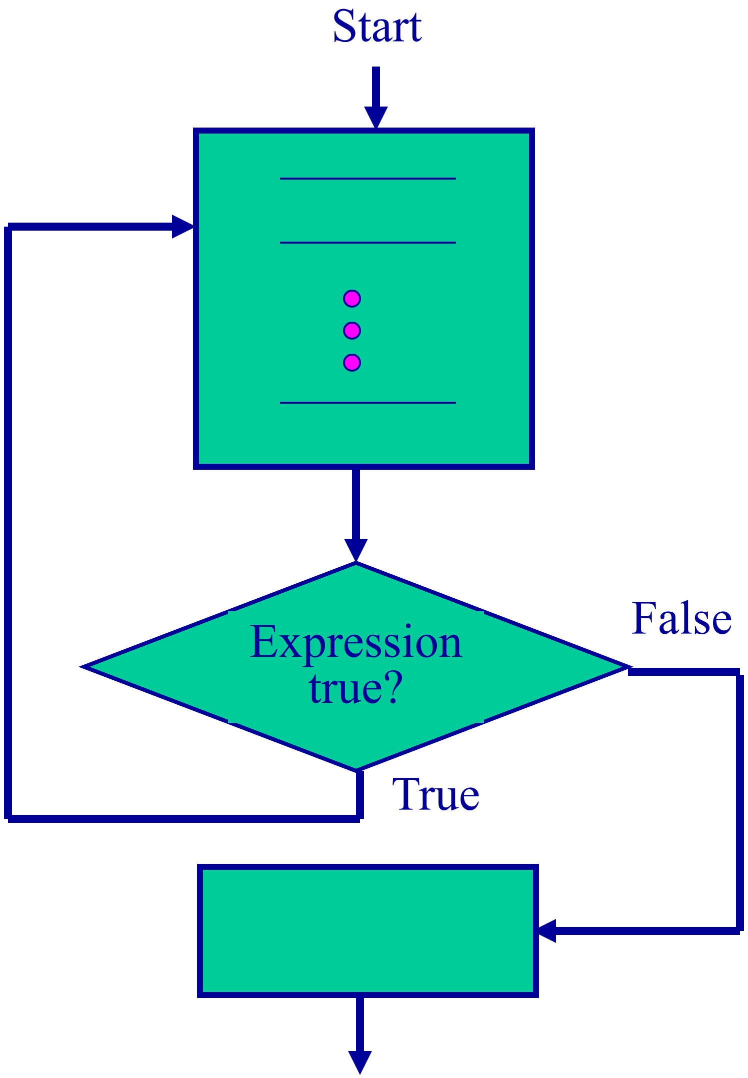
```
do

{

      action

} while ( expression ) ;
```

Note: don't forget the semicolon after the while statement.



Note:

The do while statement is similar to the while statement; the only difference is that the expression controlling the loop is tested at the bottom of the loop. So the body of the loop is always executed at least once.

The do while statement

```
#include <stdio.h>
main ()
{
     int x = 0;
     printf ( “x = %d\n”, x );
     do
    {
              x = x - 2;
              printf ( “x = %d\n”, x );
     } while ( x >= 1);
}
```

The while statement

```
#include <stdio.h>
main ()
{
     int x = 0;
     printf ( “x = %d\n”, x );
     while ( x >= 1)
    {
              x = x - 2;
              printf ( “x = %d\n”, x );
     }
}
```

the output is

```
x = 0
x = -2
```

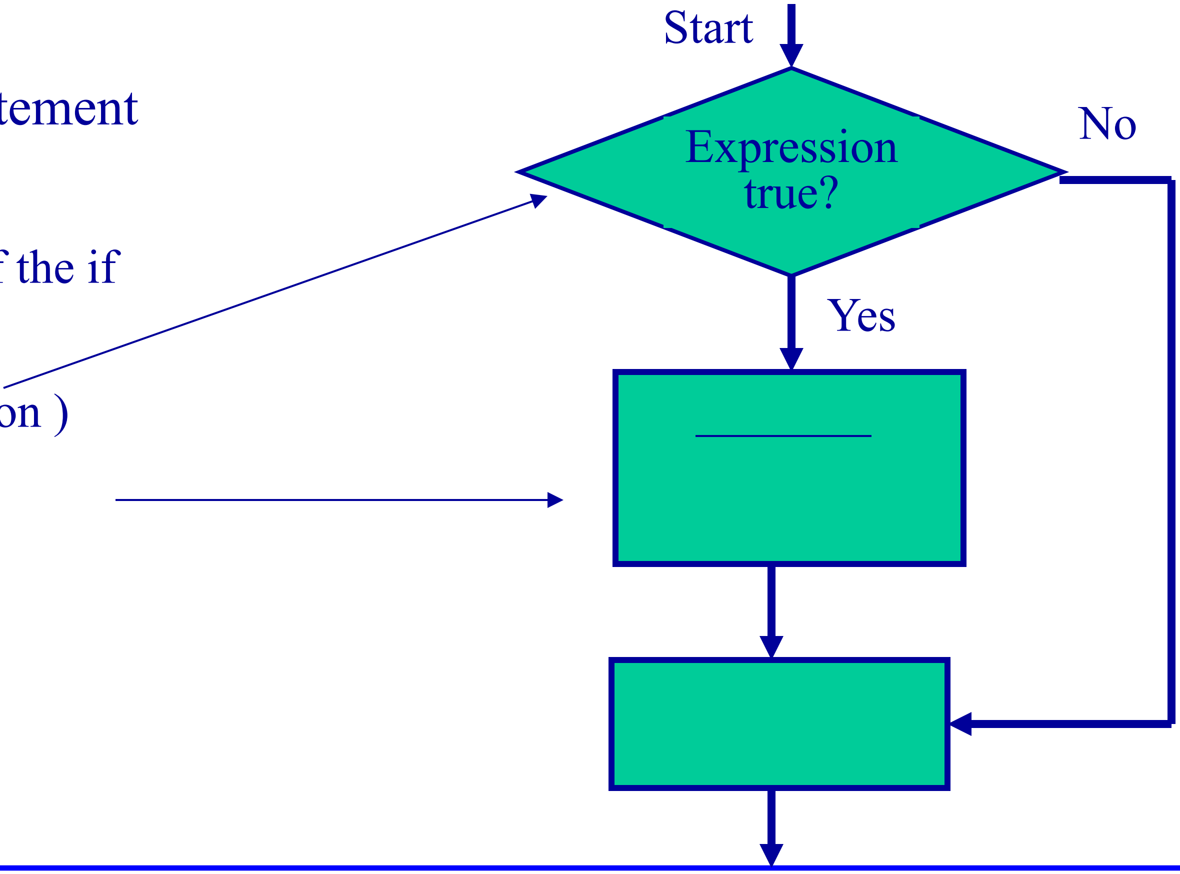
the output is

```
x = 0
```

The if Statement

One form of the if statement is

```
if ( expression )
        action
```



```
#include <stdio.h>
main ( )
{
        int code;
        code = 1;
        if (code == 1)
        {
                printf ( “Electrical and Computer Engineering\n”);
                printf ( “Professor Wright\n” );
                printf ( “Room 111\n”);
        }
        printf ( “*** End of course listing ***”);
}
```

The output is

```
Electrical and Computer Engineering
Professor Wright
Room 111
*** End of course listing ***
```

Difference between while and if

```
while (expression)
{
      action_1
}
action_2
```

Come back

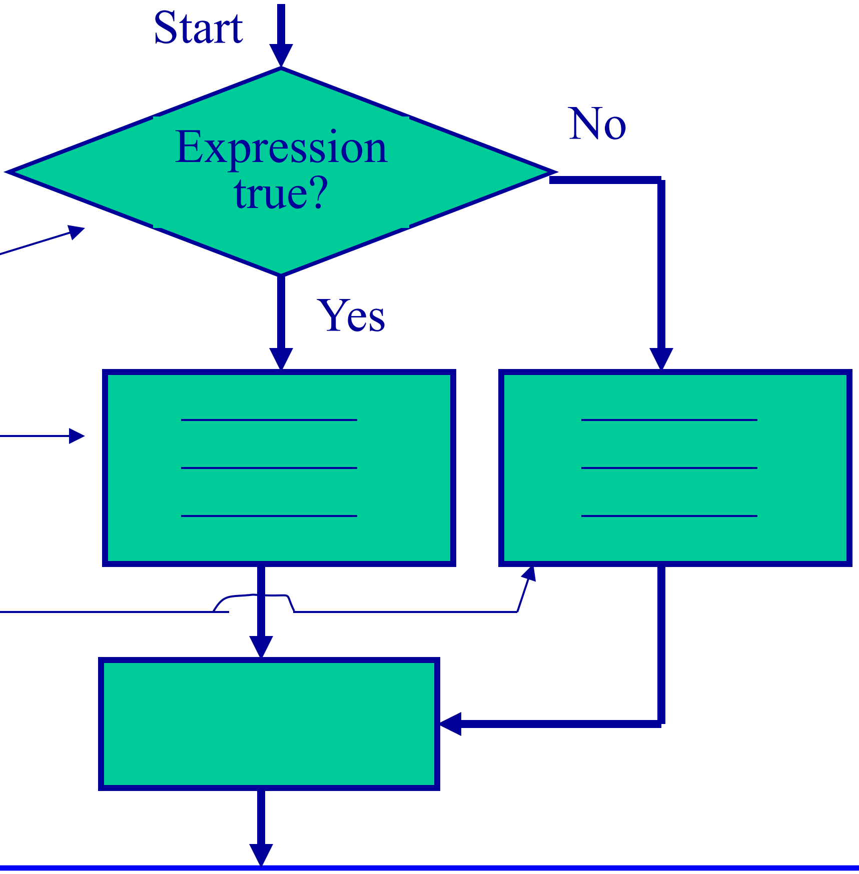
```
if (expression)
{
      action_1
}
action_2
```

leave

if…else statement

The other form of the if statement is

```
if ( expression )
        action 1
else
        action 2
```



```
#include <stdio.h>
main ( )
{
        int code;
        code = 2;
        if (code != 1)
                printf ( “No course listed\n” );
        else {
                printf ( “Electrical and Computer Engineering\n”);
                printf ( “Professor Wright\n” );
                printf ( “Room 111\n”);
        }
        printf ( “*** End of course listing ***”);
}
```

The output is

```
No course listed
*** End of course listing ***
```

Note: if or if-else can be the action of another if or if-else statement.

```
#include <stdio.h>
main ( )
{
    int code, course_code;
    code = 2;
    course_code = 1;
    if (code == 1)
        if ( course_code ==2 )
            printf ( “Mechanical Engineering\n” );
        else
            printf ( “No course listed\n” );
    else
        if (course_code == 1)
            printf ( “Electrical and Computer Engineering\n”);
        else
            printf ( “No course listed\n” );
    printf ( “*** End of course listing ***”);
}
```

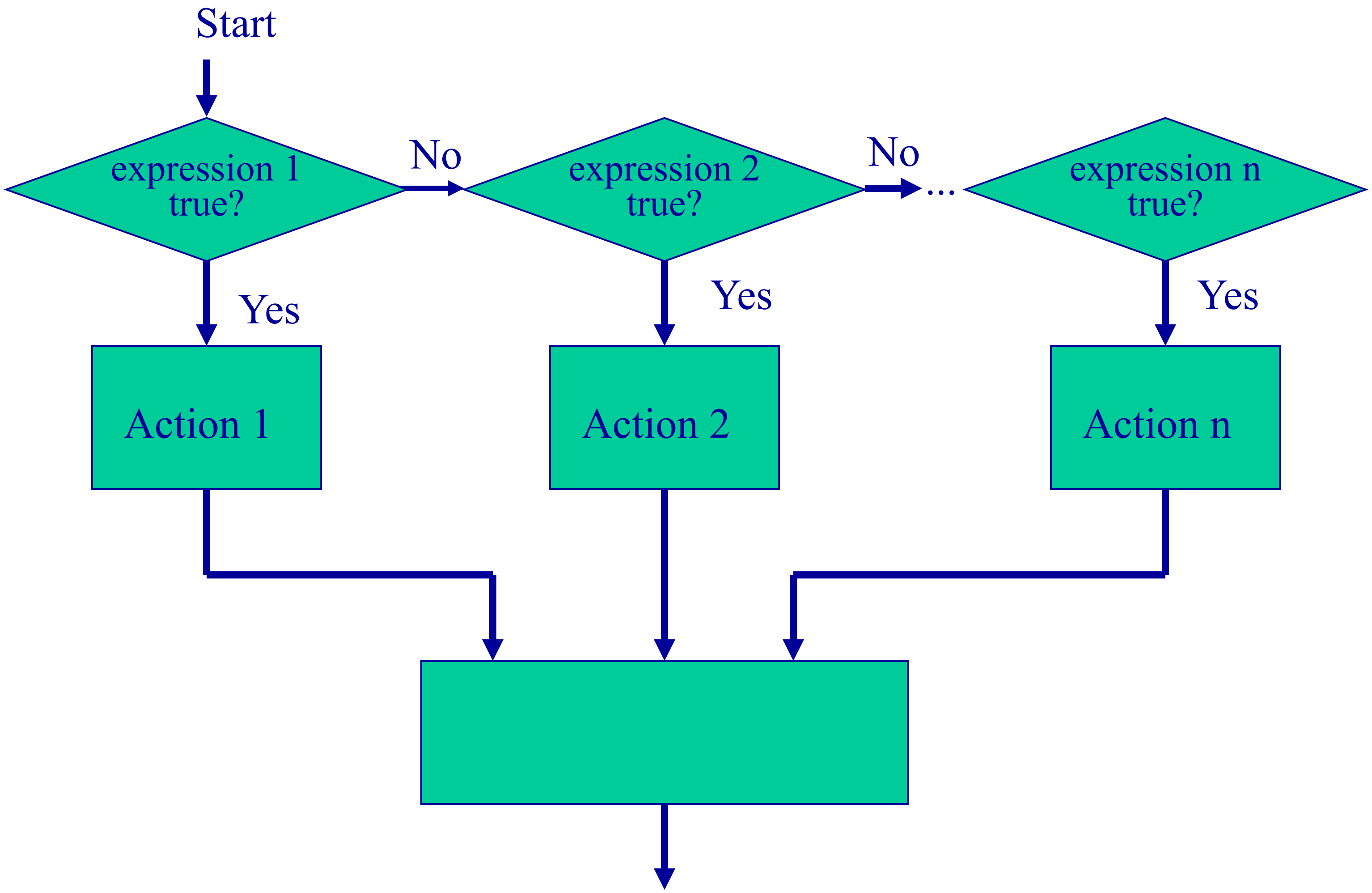
The output is

```
Electrical and Computer Engineering
*** End of course listing ***
```

More on the if Statement

Nested if-else statements are typically written

```
if ( expression 1 )
        action 1
else if ( expression 2 )
        action 2
else if ( expression 3 )
        action 3
…
else if ( expression n )
        action n
```


Start
expression 1 true?
No
expression 2 true?
No
…
expression n true?
Yes
Yes
Yes
Action 1
Action 2
Action n

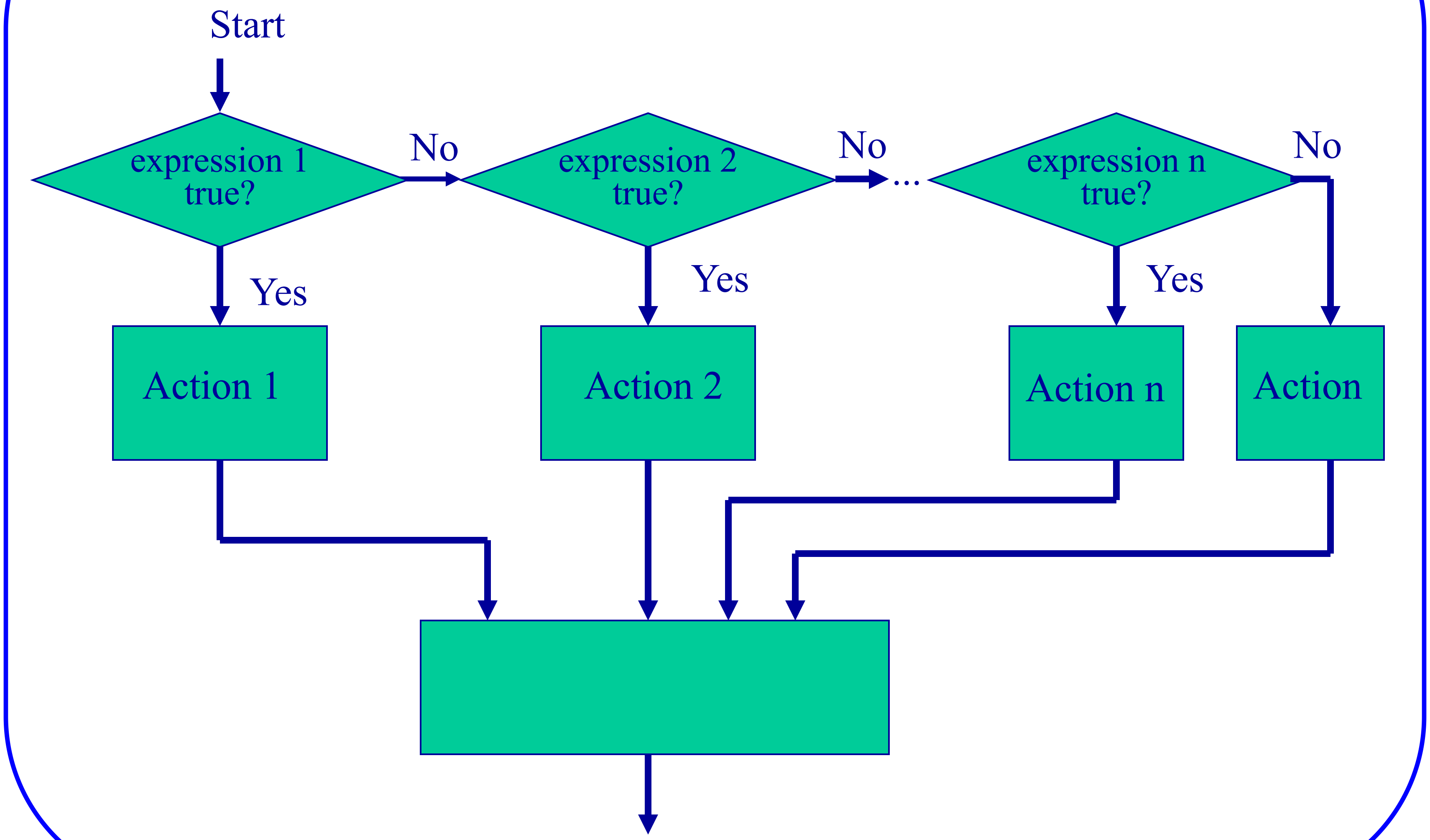
```
#include <stdio.h>
main ( )
{
        int code;
        code = 2;
        if (code == 1)
                printf ( “Mechanical Engineering\n” );
        else if ( code == 2)
                 printf ( “Electrical and Computer Engineering\n”);
        else if (code <= 3)
                printf ( “Science\n” );
        printf ( “*** End of course listing ***”);
}
```

The output is

```
Electrical and Computer Engineering
*** End of course listing ***
```

Another useful form of if-else statement is

```
if ( expression 1 )
        action 1
else if ( expression 2 )
        action 2
else if ( expression 3 )
        action 3
…
else if ( expression n )
        action n
else
        action
```


Start
expression 1 true?
No
expression 2 true?
No
…
expression n true?
No
Yes
Yes
Yes
Action 1
Action 2
Action n
Action

```
#include <stdio.h>
main ( )
{
        int code;
        code = 4;
        if (code == 1)
                printf ( “Mechanical Engineering\n” );
        else if ( code == 2)
                 printf ( “Electrical and Computer Engineering\n”);
        else if (code <= 3)
                printf ( “Science\n” );
        else
                printf ( “No course listed\n” );
        printf ( “*** End of course listing ***”);
}
```

The output is

```
No course listed
*** End of course listing ***
```