

ECE 220 Computer Systems & Programming

Variables and Control Structures



Basic I/O

#include <stdio.h>

- **printf examples**

```
printf(“%d is a prime number”, 43);  
printf(“43 + 59 in decimal is %d\n”, 43+59);  
printf(“a+b=%f\n”, a+b);  
printf(“%d+%d=%d\n”, a, b, a+b);
```

- **scanf examples**

```
scanf(“%c”, &nextchar);  
scanf(“%f”, &radius);  
scanf(“%d %d”, &length, &height);
```

Formatting option: %d, %x, %c, %s, %f, \n,

Use “**man 3 functionName**” to look up library functions

Basic I/O

#include <stdio.h>

- **printf examples**

```
printf(“%d is a prime number”, 43);  
printf(“43 + 59 in decimal is %d\n”, 43+59);  
printf(“a+b=%f\n”, a+b);  
printf(“%d+%d=%d\n”, a, b, a+b);
```

- **scanf examples**

```
scanf(“%c”, &nextchar);  
scanf(“%f”, &radius);  
scanf(“%d %d”, &length, &height);
```

This & is “magical”

Formatting option: %d, %x, %c, %s, %f, \n,

Use “**man 3 functionName**” to look up library functions

Control Structures

Conditional Constructs

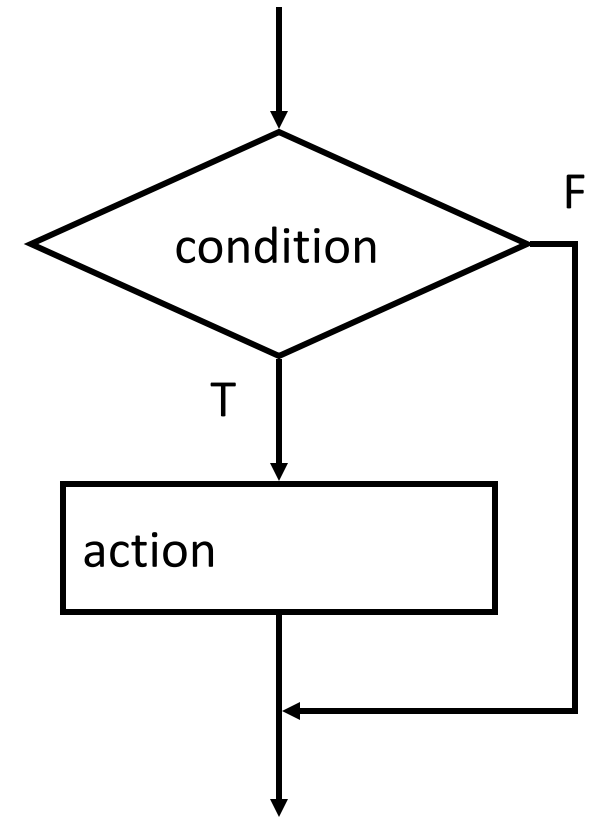
- **if**
- **if - else**
- **switch**

Iteration Constructs (loops)

- **while**
- **do - while**
- **for**

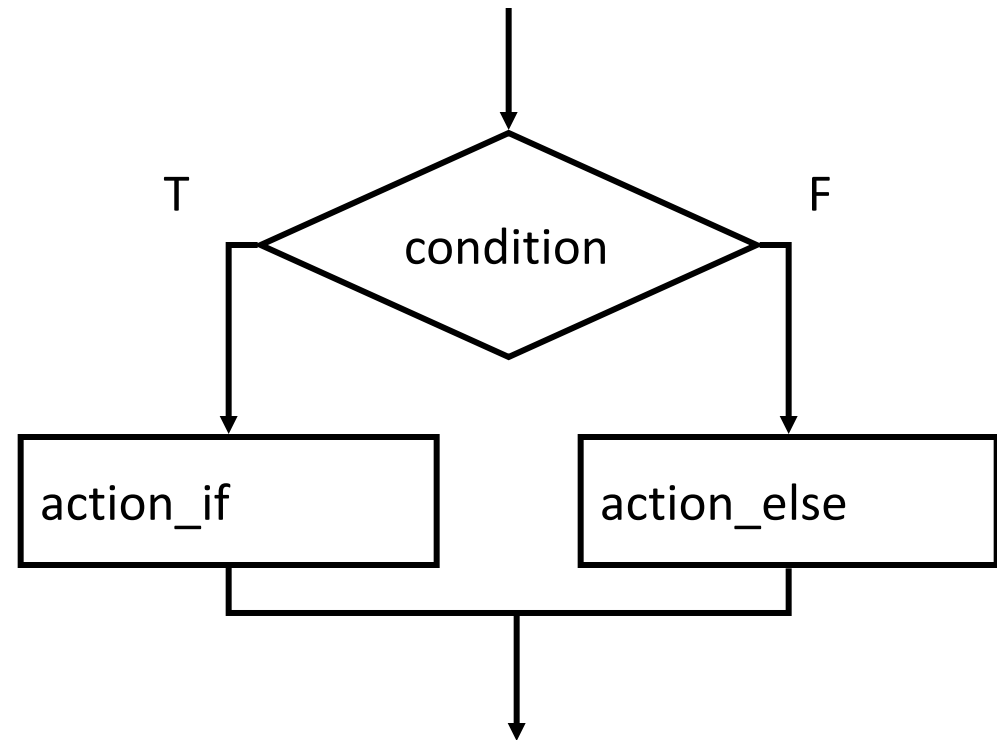
The if Statement (similar to BR in LC-3)

```
int x;  
... //assign some value to x  
if (x < 0)  
    x = -x; //invert x only if x < 0  
  
int y = 0;  
if ((x > 5) && (x < 25))  
{  
    y = x * x + 5;  
    printf("y = %d\n", y);  
}
```

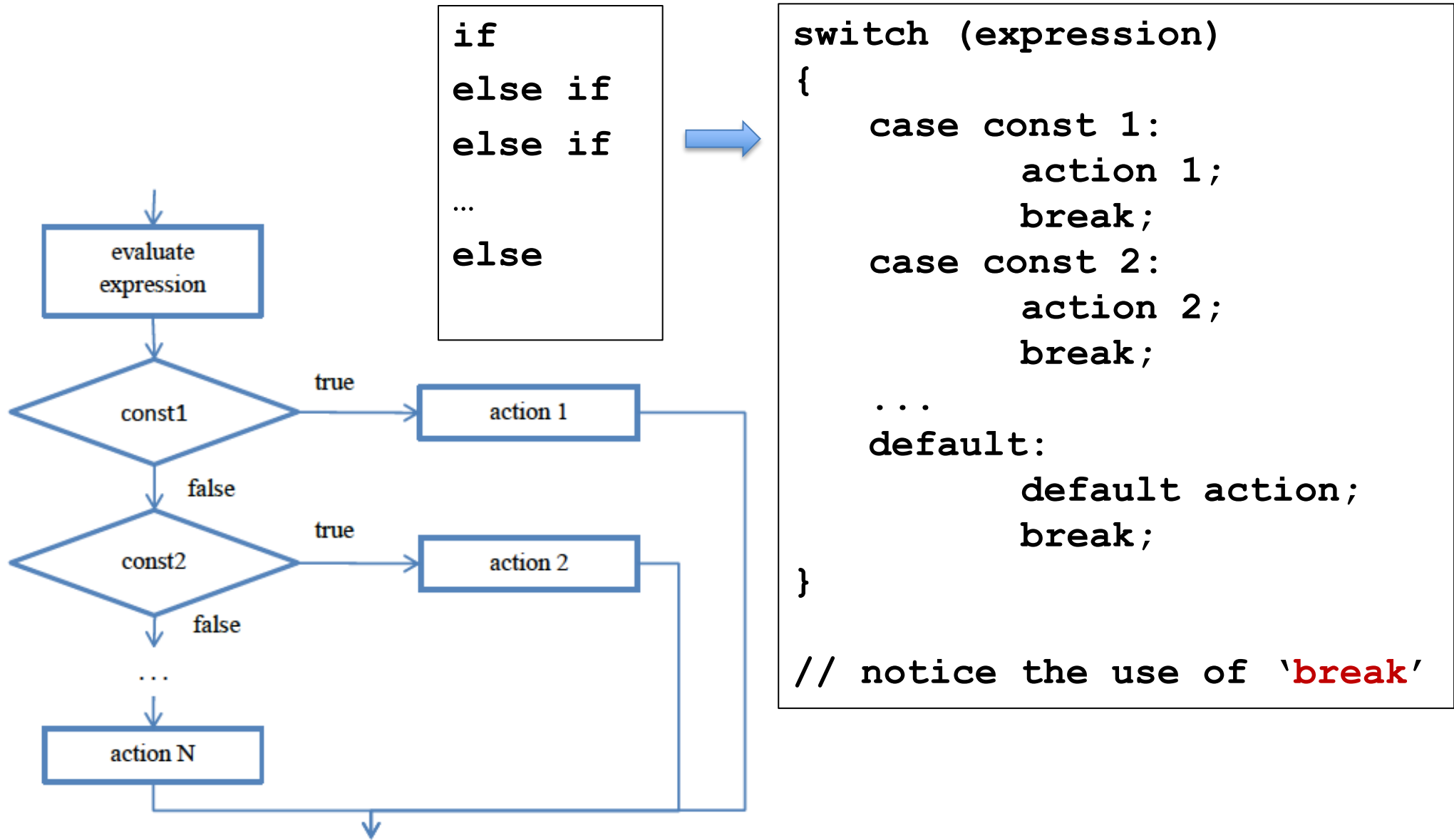


The if - else Statement

```
/*x and y are of type int*/  
if (x < 0)  
    x = -x;  
else  
    x = x * 2;  
  
if ((x > 5) && (x < 25))  
{  
    y = x * x + 5;  
    printf("y = %d\n", y);  
}  
else  
    printf("x = %d\n", x);
```

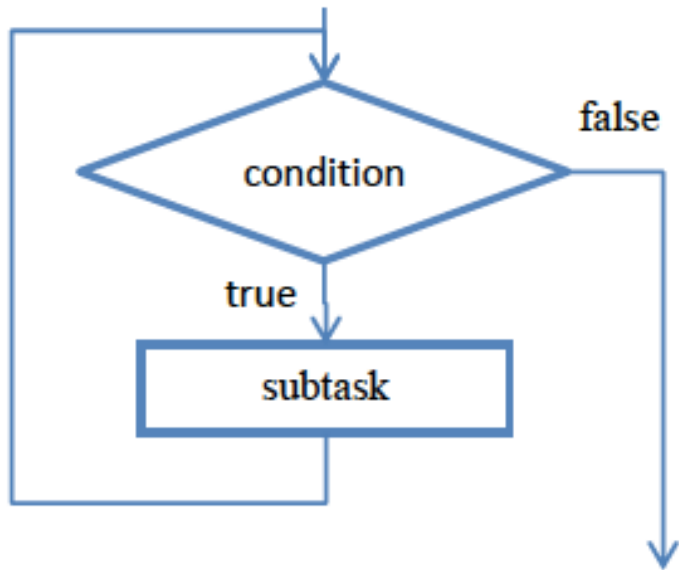


The switch Statement



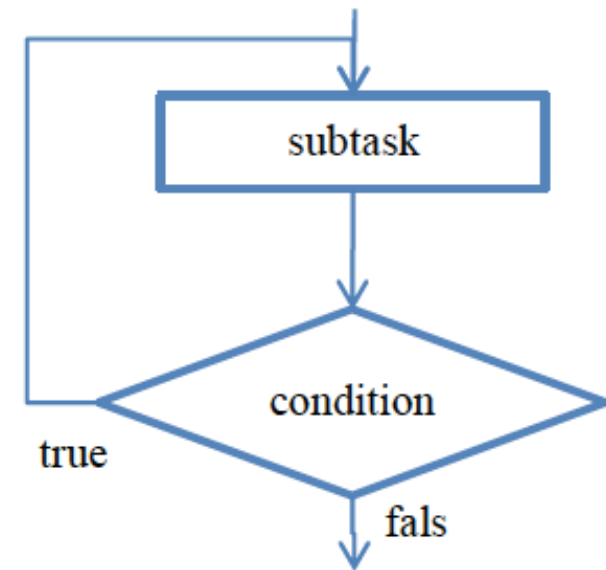
The while / do - while Statement

while: loop body may or may not be executed even once



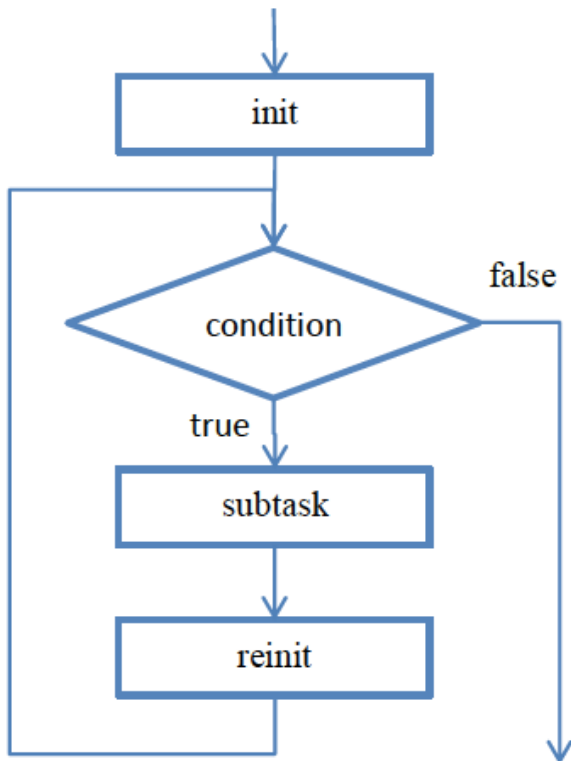
```
int x = 0;
while (x < 10) {
    printf("x=%d\n", x);
    x = x + 1;
}
```

do – while: loop body will be executed at least once



```
int x = 0;
do {
    printf("x=%d\n", x);
    x = x + 1;
} while (x < 10);
```


The for Statement



```
int x = 0;
while (x < 10) {
    printf("x=%d\n", x);
    x = x + 1;
}
```

```
int x;
for (x = 0; x < 10; x++)
{
    printf("x=%d\n", x);
}
```

➤ What would cause while loop or for loop to become infinite loops?

```
for (x = 0; x < 10; x++) {
    if (x == 5)
        break;
    printf("x=%d\n", x);
} /* what would be the print out? */
```

break and continue

- break: leave the loop
- continue: enter next iteration:
 - while and do-while: jump to test
 - for: jump to *re-init*

Nested Loops

inner loop is nested within the outer loop (similar to print hex example in LC-3)

```
for ()
{
    for ()
    {
        ...
    } // inner loop to shift 4 bits to calculate each digit
    ...
} // outer loop to print the 4 digits
```