C Programming & Lab

6. Loops

Sejong University

Outline

1) **Loops**
2) **while loop**
3) **do while loop**
4) **for loop**
5) **Nested Loops**
6) **break statement**
7) **continue statement**

1) Selection Statements

- **Control Statement**
  - C programming language sequentially processes the source code from the first line (if not specifically stated)
  - May need to change the linear flow of the source code → Control Statements

- **Types**

Type	Examples
Selection Statements	if statement, if~else statement, switch statement
Loop Statements	**for statement, while statement, do~while statement**
Others	**break statement, continue statement,** goto statement, return statement

1) Loops

- **Loops**
  - Need to repeat a block of code until a particular condition is satisfied
  - while loop, do while loop, for loop

  - Example

```
// Repeat the same statement
printf(“Hello World\n”);
printf(“Hello World\n”);
printf(“Hello World\n”);

// Repeat a job with a particular pattern
int sum = 1 + 2 + 3 + 4 + 5;
```

2) while loop

- **If (controlling) *expression* is true, execute *statement* repeatedly.**
- **Terminate the loop when the (controlling) statement becomes false.**
  - Typically, use increment/decrement statement.

- **Syntax**

```
while( expression ) {
    statement;
    increment/decrement;
}
```

  - Example) Print "Hello World" 3 times

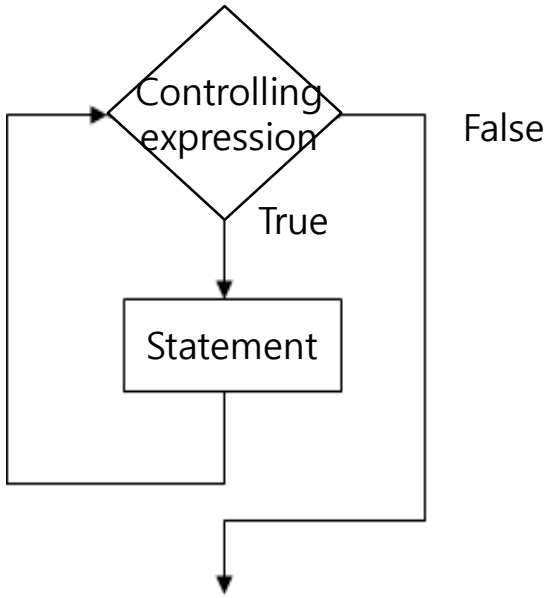
```
while( i <= 3 ) {
    printf( " Hello World\n");
    i++;
}
```

2) while loop

- **Use 'while loop' and 'do while loop' when you know the condition (controlling expression) but not the number of iteration**
  - while loop: the *controlling expression* is evaluated first
  - do while loop: Execute the *statement* first
- **(for loop: when a number of iterations is fixed)**

- **while loop**

```
while( expression ) {
      statement;
}
```



2) while loop

- **Use while loop to calculate the sum of 1 to 10**

```
/* Use while loop to calculate the sum of 1 to 10 */

#include <stdio.h>

  int main(void) {
   int i, sum;
   i = 1;
   sum = 0;

   while ( i <= 10) {
        sum += i;
        i++;
  }

  printf("Sum of 1 to 10 is %d\n", sum);
  return 0;
}
```

2) while loop

- **Use while loop to calculate the sum of integers that are received until a character is received**

```
#include <stdio.h>

int main(void) {
   int i, total;
   i = 0;
   total = 0;

   prinft(“Enter an integer: ”);
   while (scanf(“%d”, &i)) {
        total += i;
        prinft(“Enter an integer: ”);
   }

   printf(“Sum of the received integers is %d\n”, total);
   return 0;
}
```

2) while loop

- **Write a while loop**

1. If a variable *i* is less than 10, print "Hello World" and increase *i* by 1

2. If a variable *i* is less than 10, assign 0 to a variable *x* and decrease *i* by 1

3. Keep reading a number (variable *num*) as long as it is less than 50.

2) while loop (Example)

- **(Example) Read an (positive) integer *n*, print "Hello World" *n* times.**

Input Example

```
3
```

Output Example

```
Hello World
Hello World
Hello World
```

- **(Example) Print alphabet letters (Upper-case, Lower-case).**

Alphabet	Decimal
A	65
B	66
Z	90
a	97
b	98
z	122

Output Example

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
```

2) while loop (Example)

- **(Example) What is the number *a* that the sum of the first *a* numbers from 1 exceeds 100?**

Output Example

```
14
```

- **(Example) Read an integer *n*, print the multiplication table for *n*.**

Input Example

```
5
```

Output Example

```
5 * 1 = 5
5 * 2 = 10
…
5 * 9 = 45
```

2) while loop (Example)

- **(Example5) Read four integers, print the smallest integer.**

Input Example

```
10 2 15 7
```

Output Example

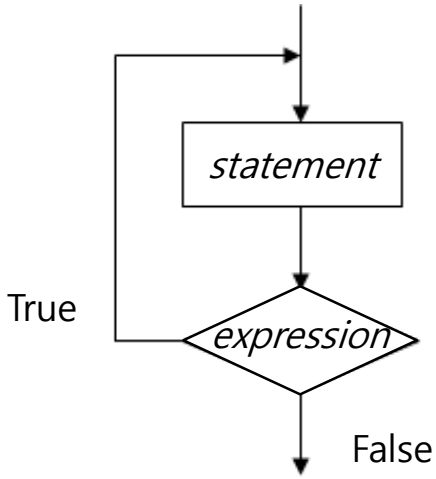
```
2
```

3) do while loop

- **Execute *statement* (do) first, evaluate the (controlling) *expression***
- **Execute the *statement* at least once**
- **Do not forgot to add semi-colon (;)**

- **Syntax**

```
do {
    statement;
} while( expression );
```



3) do while loop

- **Use while loop to calculate the sum of 1 to 10**

```
#include <stdio.h>

int main(void) {
   int i, sum;
   i = 1;
   sum = 0;

  do {
        sum += i;
        i++;
  } while ( i <= 10) ;

 printf("Sum of 1 to 10 is %d\n", sum);
 return 0;
}
```

3) do while loop

- **Use while loop to calculate the sum of integers that are received until a character is received**

```
#include <stdio.h>

int main(void) {
   int i, total;
   i = 0;
   total = 0;

do {
      prinft(“Enter integer: ”);
      total += i;
   } while (scanf(“%d”, &i));

   printf(“Sum of the received integers is %d\n”, total);
   return 0;
}
```

3) do while loop

- **Print the number repetition using do while loop**

```
#include <stdio.h>

int main(void)
{
	int count = 1;
	do {
		printf("do while loop repetition: %d\n", count);
		count++;
	} while(count < 5);
	printf("number of repetition:%d\n", count - 1);

}
```

3) do while loop (Example)

- **(Example) Read an integer *n*, print "Hello World" *n* times.**

Input Example

```
3
```

Output Example

```
Hello World
Hello World
Hello World
```

- **(Example) Read an integer *n*, print the multiplication table for *n*.**

Input Example

```
5
```

Output Example

```
5 * 1 = 5
5 * 2 = 10
…
5 * 9 = 45
```

2) do while loop (Example)

- **(Example) What is the number *a* that the sum of the first *a* numbers from 1 exceeds 100?**

Output Example

```
14
```

- **(Example) What is the largest number *a* that the sum of the first *a* numbers from 1 does not exceed? Print the sum of the first *a* numbers as well.**

Output Example

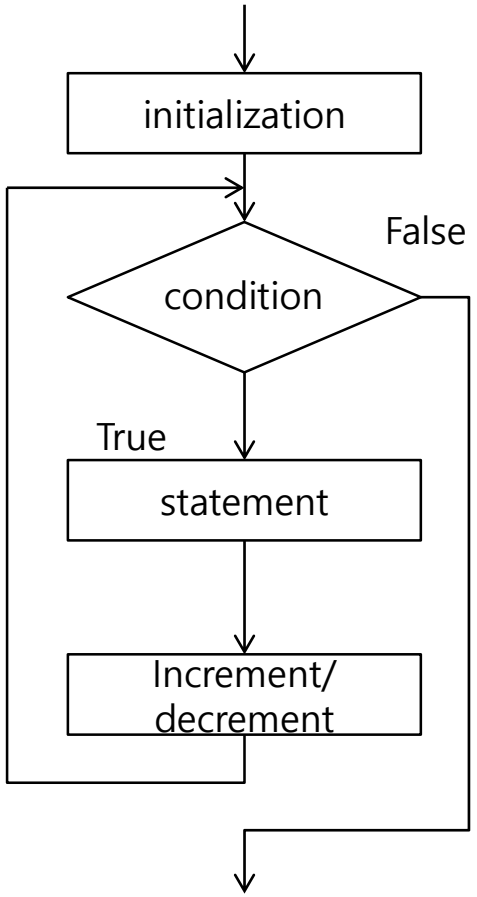
```
44
990
```

4) for loop

- **The number of repetition is fixed**
- **Evaluate first whether it executes the *statement* or not**
- **initialization, condition, increment/decrement**

- **Syntax**

```
for(initialization; condition; increment/decrement) {
      statement;
}
```



4) for loop

- **Initialization**
  - Execute once
  - Use comma(,) to list multiple statements.

```
for (i=0, sum=0; i<=10 ; i++)
```

  - May leave as blank

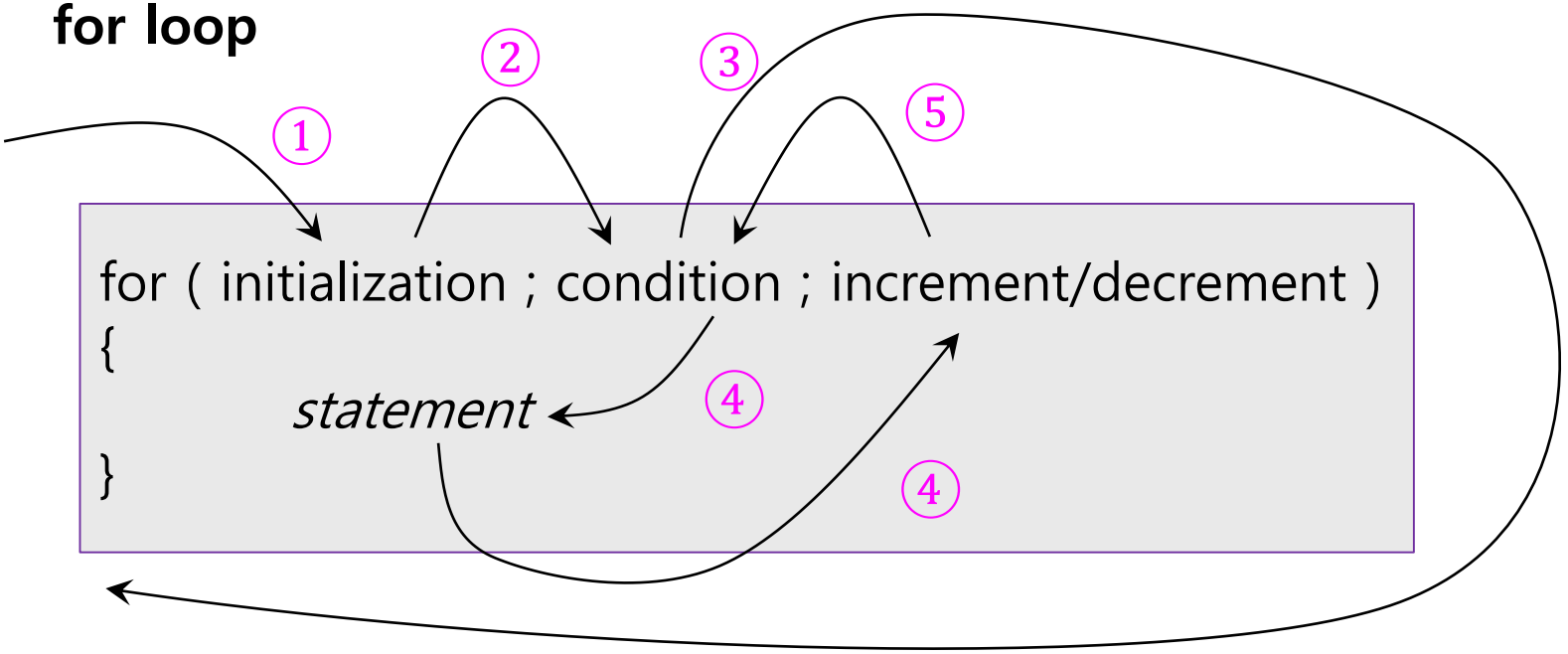
- **Condition**
  - Repeat until the condition is true
  - Evaluate it first
  - Use only logical variable or expression

```
for (i=0; i<=10 && sum<100 ; i++)
```

- **Increment/decrement**
  - Execute the statement and then execute increment/decrement
  - Increase/decrease a variable

4) for loop

- **for loop**



① Initialize
② Evaluate the condition
③ If false, terminate for loop
④ If true, execute *statement*
⑤ Increment/decrement, go to ②

4) for loop

- **Write a for loop**

  1. Initialize a variable *i* of integer type with 0, increase *i* by 1, iterate the loop until *i* reaches 10
  2. Initialize a variable *i* of integer type with 10, decrease *i* by 3, iterate the loop until *i* is greater than or equal to 0
  3. Initialize a variable *i* of integer type with 0, increase *i* by *i*i+2*, iterate the loop until *i* is smaller than 50

4) for loop

- **Use for loop to calculate the sum of 1 to 10**

```
#include <stdio.h>

int main(void) {
   int i, sum;
   sum = 0;

   for(i=1; i<=10 ; i++)
        sum += i;

 printf("Sum of 1 to 10 is %d\n", sum);
 return 0;
}
```

4) for loop

- **Calculate factorial by using for loop**

```
/* Calculate 5! using for loop */

#include <stdio.h>

int main(void) {
   int i, fac;
   fac = 1;

   for(i=1; i<=5; i++) {
      fac = fac * i;

   printf("5! is %d\n", fac);
   return 0;
}
```

4) for loop

- **Comparison: while loop vs. for loop**

```
// while loop
①
Initialization;
②
 while( expression ) {
    ③statement;
    ④increment/decrement;
 }
```

```
// for loop
①              ②                    ④
 for(initialization; condition; increment/decrement) {
    ③ statement;
}
```

4) for loop

- **Infinite loop**
  - Loop that never terminates

```
while( 1 ) {
      statement;
}
```

```
for( ; ; ) {
      statement;
}
```

  - Use *break statement* to terminate the loop
  - Use shift+F5 to terminate debugging

4) for loop

- **Convert wile loop to for loop**

1.
```
for(int i=1; i<5; i++)
     printf(“%d”, i);
```

2.
```
for(i=10; i>=0; i-=2)
     printf(“%d”, i);
```

- **Convert wile loop to for loop**

1.
```
int i=5;
while(i>=0){
      printf(“%d”, i);
      i--;
}
```

2.
```
int i=0;
while(i<=50){
      if(i%5==0)
         printf(“%d”, i);
      i+=2;
}
```

4) for loop (Practice)

- **(Practice) Read an integer and compute factorial of the integer**

Input Example

```
5
```

Output Example

```
120
```

- **(Practice) Read a character *c* and integer *n* and print the character *n* times**

Input Example

```
A 5
```

Output Example

```
A A A A A
```

4) for loop (Practice)

- **(Practice) Compute the sum of odd and even numbers from 1 to 10 using for loop**

Output Example

```
25
30
```

- **(Practice) Read an integer *n* and print the calendar of the *n*th month of this year**

Input Example

```
5
```

Output Example

```
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
...
```

5) Nested loop

- **Use a loop statement inside another loop statement**
- **while loop inside another while loop**
- **for loop inside another for loop**
- **while loop and for loop can be used together**

- **Nested for loop**

```
for(initialization; condition; increment/decrement) {
      statement1;
       for(initialization; condition; increment/decrement) {
            statement2;
       }
      statement3;
}
```

5) Nested loop

```
/* Two for loops to print multiplication tables */

int  j, k;
 for( j = 2;  j < 10;  j++)
    for(k = 1;  k < 10;  k++)
       printf ("%d * %d = %d\n", j, k, j * k);
```

```
/* while loop, for loop to print multiplication tables */

int  j, k;
j = 2;
while( j < 10 )  {
     for (k = 1;  k < 10;  k++)
         printf ("%d * %d = %d\n", j, k, j * k);
j++;
```

5) Nested loop (Practice)

- **(Practice) Use nested for loop to print the example 1 and 2**

Output Example 1

```
*
**
***
****
*****
```

Output Example 2

```
 1  2  3  4  5
 6  7  8  9
10 11 12
13 14
15
```

- **(Practice) Use nested for loop to print prime numbers between 2 and 50**

Output Example

```
2  3  5  7  11
13  17  19  23  29
31  37  41  43  47
```

6) break statement

- ***Break*** **and** ***continue*** **statements to control loops**

- **Terminate loops regardless of the loop condition**

6) break statement

- **break statement**

```
 while( expression ) {
     statement;
     break;
     statement 1;
 }
statement 2;
```

```
do {
      statement;
      break;
      statement 1;
while( expression );
statement 2;
```

```
for(initialization; condition; increment/decrement) {
      statement;
      break;
      statement 1
}
statement 2;
```

6) break statement

- **break statement example**

```
#include <stdio.h>

int main()
{
   int count;
   for (count = 0; count < 5; count++) {
       if (count == 3)
       break;
    }
   printf(“count is %d\n”, count);
}
```

6) break statement (Practice)

- **(Practice) What is the number *a* that the sum of the first *a* numbers from 1 exceeds 100? Use infinite loop and break statement**

Output Example

```
14

105
```

- **(Practice) Read integers. Calculate and print the sum and average of the integers. Use infinite loop and break statement. Terminate the loop if it receives 0**

Input Example

```
1

5

3

0
```

Outpu Example

```
9

3
```

7) continue statement

- **Stop the current iteration and evaluate the next condition**

- **Go to the end of the loop and start the next iteration**
  - Do not execute statements following *continue statement*

7) continue statement

- **continue statement**

```
 while( expression ) {
     statement;
     continue;
     statement 1;
 }
statement 2;
```

```
do {
      statement;
      continue;
      statement 1;
while( expression );
statement 2;
```

```
for(initialization; condition; increment/decrement) {
      statement;
      continue;
      statement 1
}
statement 2;
```

7) continue statement

- **Use continue statement, print even numbers between 1 and 10**

```
#include <stdio.h>

int main()
{
    int i;
    printf(“ Print even numbers between 1 and 10\n”);

    for (i = 0; i <= 10; i++) {
        if (i % 2 != 0)
           continue;
        printf(“\n%d”, i);
     }
}
```

7) continue statement (Practice)

- **(Practice) Calculate the sum of odd numbers using continue statement**

Output Example

```
25
```

- **(Practice) Print numbers between 1 and 20 except the numbers that are both multiple of 2 and 3**
  - Hint:
    - ✓ Multiple of 2 (i%2)==0
    - ✓ Multiple of 3 (i%3)==0

Output Example

```
1 2 3 4 5 7 8 9 10
```