

3강. 제어문

C++ 프로그래밍

jhhwang@kumoh.ac.kr

목차

- ▶ 제어문
- ▶ 선택문
 - if 문
 - swich 문
- ▶ 반복문
 - for 문
 - while 문
 - do ~ while 문
- ▶ 프로그래밍 문제

제어문

▶ 제어문

- 실행 흐름을 변경시켜 주는 문장

선택문	반복문	분기문
특정 조건에 따라 실행 여부 선택	특정 부분을 반복적으로 수행	특정 위치로 이동
if switch	for while do ~ while	break continue return goto

선택문 : if 문

▶ 문법

```
if (condition)  
    statement
```

```
if (a < 5)  
    a = 100;
```

```
if (condition)  
    statement  
else  
    statement
```

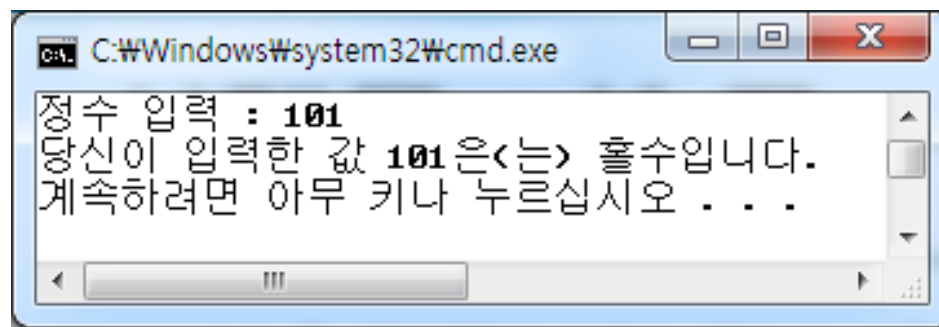
```
if (condition) {  
    statements  
}
```

```
if (condition) {  
    statements  
}  
else {  
    statements  
}
```

```
if (a < 3)  
    cout << "a는 3보다 작다." << endl;  
else {  
    cout << "a는 3보다 작지 않다." << endl;  
    a = 0;  
}
```

선택문 : if 문 – 연습 문제

- ▶ 사용자로부터 정수 하나를 입력 받아 짝수이면 “짝수”, 홀수이면 “홀수”라고 출력하는 프로그램을 작성하라.



선택문 : if 문 – 연습 문제

▶ 프로그램 확인

```
#include <iostream>
using namespace std;

void main(void)
{
    int Num;

    cout << "정수 입력 : ";
    cin >> Num;

    cout << "당신이 입력한 값 " << Num << "은(는)";
    if (Num % 2 == 0)
        cout << "짝수입니다." << endl;
    else
        cout << "홀수입니다." << endl;
}
```

선택문 : switch 문

▶ 문법

정수형(int, char, bool)만 허용

```
switch (condition) {  
  case val1 :  
    statements;  
  case val2 :  
    statements;  
  .....  
  default :  
    statements  
}
```

```
switch (condition) {  
  case val1 :  
    statements;  
    break;  
  case val2 :  
    statements;  
    break;  
  .....  
  default :  
    statements  
    break;  
}
```

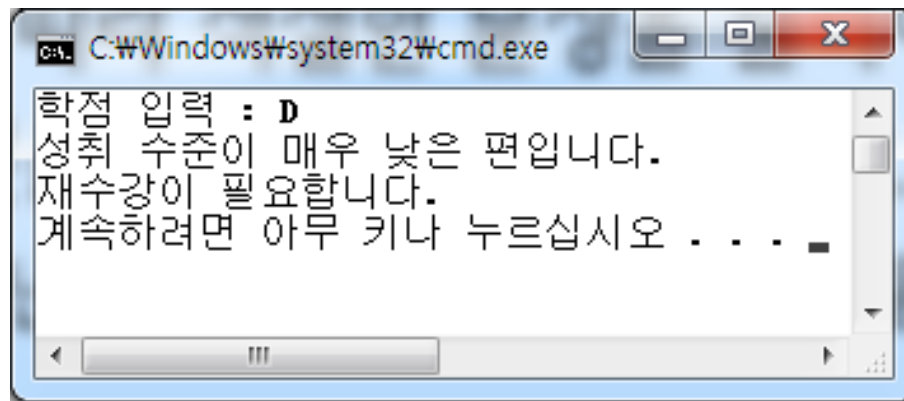
switch 문은 if 문의
부분 집합

```
int age = 21;
```

```
switch (age) {  
  case 20 :  
    cout << "20세" << endl;  
    break;  
  case 21 :  
    cout << "21세" << endl;  
    break;  
  case 22 :  
    cout << "22세" << endl;  
    break;  
  default :  
    cout << "20세보다 작거나 "  
          "22세보다 크다" << endl;  
    break;  
}
```

선택문 : switch 문 – 연습 문제

- ▶ 사용자로부터 자신의 학점('A', 'B', 'C', 'D', 'F')을 입력 받고 학점에 따라 적절한 문장을 출력하는 프로그램을 switch 문을 사용하여 만들어 보라.
 - 학점이 아닌 문자라면 입력이 잘못되었음을 출력해야 한다.



선택문 : switch 문 – 연습 문제

▶ 프로그램 확인

```
#include <iostream>
using namespace std;

void main(void)
{
    char Score;

    cout << "학점 입력 : ";
    cin >> Score;
```

```
switch (Score) {
case 'A' :
    cout << "매우 잘했습니다." << endl;
    break;
case 'B' :
    cout << "잘했습니다." << endl;
    break;
case 'C' :
    cout << "조금 더 분발하세요." << endl;
    break;
case 'D' :
    cout << "성취 수준이 매우 낮은 편입니다." << endl;
    cout << "재수강이 필요합니다." << endl;
    break;
case 'F' :
    cout << "공부를 다시 해야겠군요." << endl;
    break;
default :
    cout << "입력이 잘못 되었습니다.";
    break;
}
```

반복문 : for 문

▶ 문법

변수 초기화 문장

참이 되면 statement 실행

변수값 변경

```
for (statement1; condition; statement2)
    statement
```

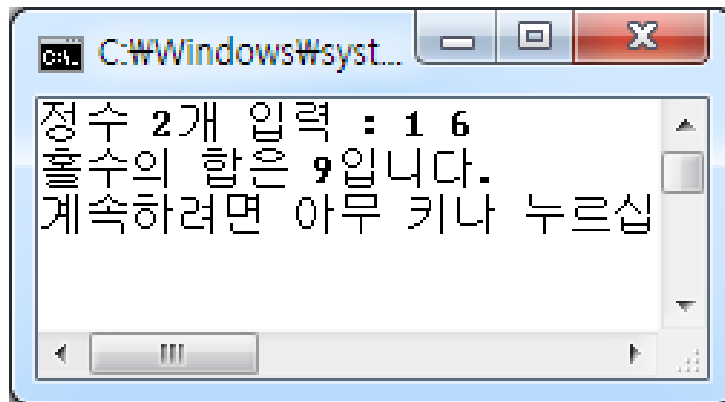
```
for (statement1; condition; statement2) {
    statements
}
```

```
int Sum = 0;

for (int i = 1; i <= 10; i++) {
    Sum = Sum + i;
}
```

반복문 : for 문 – 연습 문제

- ▶ 사용자로부터 2개의 정수를 입력받아 그 사이의 값들 중 홀수만을 더한 결과를 출력하는 프로그램을 작성하라.
 - 사용자 입력은 첫 번째 수가 두 번째 수보다 작다고 가정한다.



반복문 : for 문 – 연습 문제

▶ 프로그램 확인

```
#include <iostream>
using namespace std;

void main(void)
{
    int Num1, Num2;
    int Sum = 0;

    cout << "정수 2개 입력 : ";
    cin >> Num1 >> Num2;

    for (int i = Num1; i <= Num2; i++) {
        if (i % 2 == 1) {
            Sum += i;
        }
    }

    cout << "홀수의 합은 " << Sum << "입니다." << endl;
}
```

반복문 : while 문

▶ 문법

```
while (condition)  
    statement
```

```
while (condition) {  
    statements  
}
```

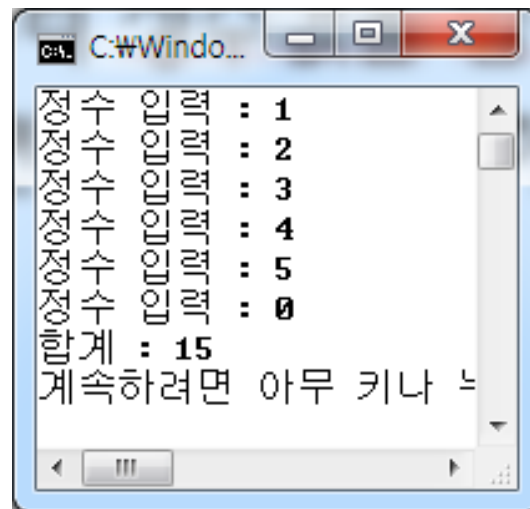
```
int Sum = 0;  
int i = 1;  
  
while (i <= 10) {  
    Sum += i;  
    i++;  
}
```

for 문에서 statement1과
statement2를 생략한 것과 동일

```
for (int i = 1; i <= 10; i++) {  
    Sum = Sum + i;  
}
```

반복문 : while 문 – 연습 문제

- ▶ 사용자로부터 정수값을 입력받아 합을 구하되 사용자가 0을 입력하면 종료하도록 하라.



반복문 : while 문 – 연습 문제

▶ 프로그램 확인

```
#include <iostream>
using namespace std;

void main(void)
{
    int Num;
    int Sum = 0;

    cout << "정수 입력 : ";
    cin >> Num;

    while (Num != 0) {
        Sum += Num;
        cout << "정수 입력 : ";
        cin >> Num;
    }

    cout << "합계 : " << Sum << endl;
}
```

반복문 : do ~ while 문

▶ 문법

반드시 1회 이상 실행

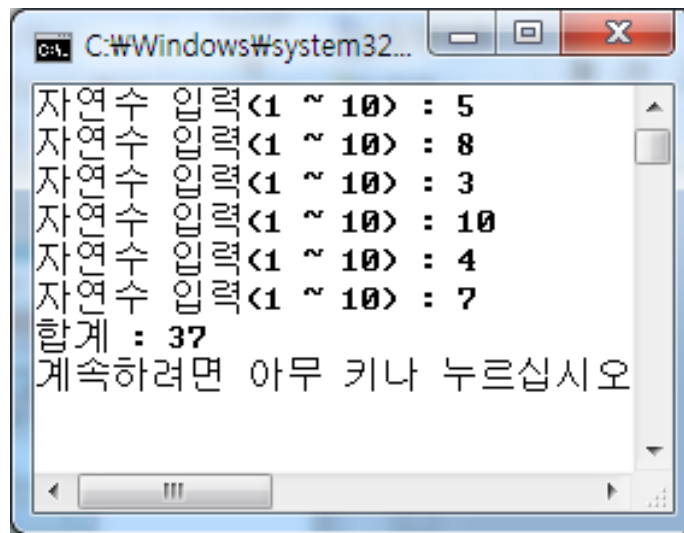
do
statement
while (condition);

do {
statements
} **while** (condition);

```
int Sum = 0;  
int i = 1;  
  
do {  
    Sum += i;  
    i++;  
} while (i <= 10);
```

반복문 : do ~ while 문 – 연습 문제

- ▶ 사용자로부터 1부터 10 사이의 임의의 값을 반복적으로 입력받되 그 값들의 합이 30을 넘으면 루프를 빠져나와 그때까지의 합을 출력하라.
 - do ~ while 문을 사용하여 구현해 보라.



반복문 : do ~ while 문 – 연습 문제

▶ 프로그램 확인

```
#include <iostream>
using namespace std;

void main(void)
{
    int Num;
    int Sum = 0;

    do {
        cout << "자연수 입력(1 ~ 10): ";
        cin >> Num;
        Sum += Num;
    } while (Sum <= 30);

    cout << "합계 : " << Sum << endl;
}
```

프로그래밍 문제

- ▶ 2개의 자연수를 입력값으로 받아 두 수의 최대공약수를 구하는 함수 **GCM**을 만들어 보라.
 - 다음 알고리즘을 참고하여 만들어 보라. (유클리드 호제법)

단계 1. 두 입력 값 중 큰 값은 M , 작은 값은 N 에 지정한다.

단계 2. M 을 N 으로 나누고 그 나머지를 R 이라고 부른다.

단계 3. R 이 0이 아닐 경우, N 이 가진 값을 M 에 지정하고

R 의 값을 N 에 지정한 다음 단계 2로 돌아간다. R 이

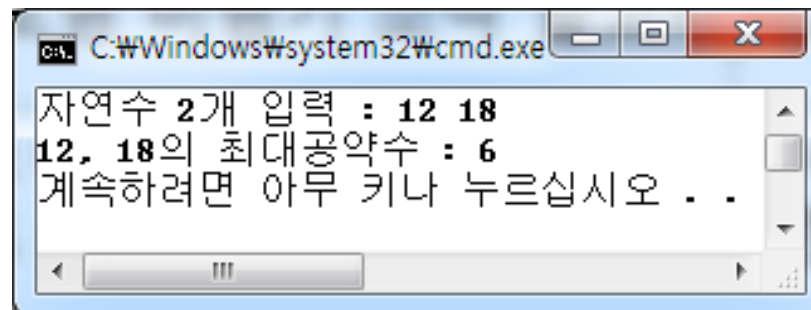
0일 경우, 현재 N 에 지정된 값이 최대공약수이다.

프로그래밍 문제

▶ 실행 예

```
void main(void)
{
    int Num1, Num2;
    cout << "자연수 2개 입력 : ";
    cin >> Num1 >> Num2;

    cout << Num1 << ", " << Num2 << "의 최대공약수 : " <<
        GCM(Num1, Num2) << endl;
}
```



프로그래밍 문제

▶ 프로그램 확인

단계 1. 두 입력 값 중 큰 값은 M, 작은 값은 N에 지정한다.

단계 2. M을 N으로 나누고 그 나머지를 R이라고 부른다.

단계 3. R이 0이 아닐 경우, N이 가진 값을 M에 지정하고 R의 값을 N에 지정한 다음 단계 2로 돌아간다. R이 0일 경우, 현재 N에 지정된 값이 최대공약수이다.

```
int GCM(int Num1, int Num2)
{
    int M, N;

    if (Num1 > Num2) {
        M = Num1;
        N = Num2;
    }
    else {
        M = Num2;
        N = Num1;
    }

    int R = M % N;

    while (R != 0) {
        M = N;
        N = R;

        R = M % N;
    }

    return N;
}
```