

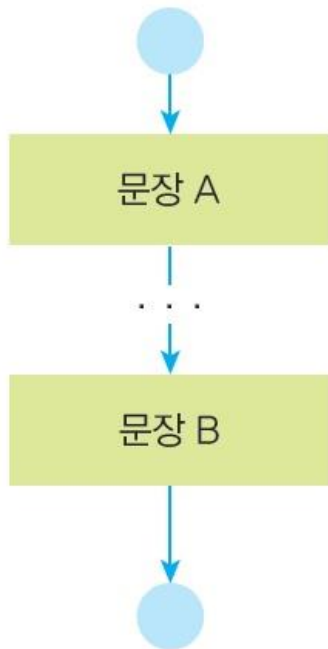
제어구조와 배열

이번 장에서 만들어볼 프로그램

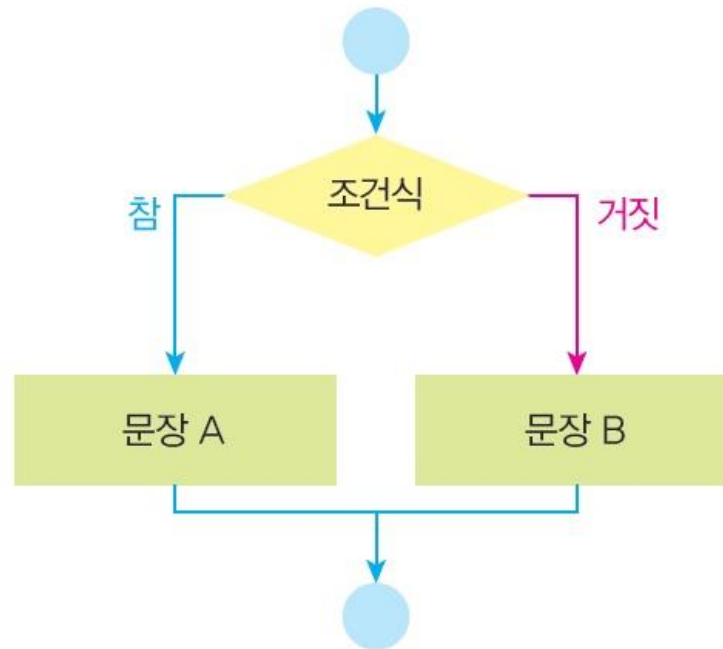
```
C:\Windows\system32\cmd.exe
(x, y) 좌표: 1 1
---|---|---
|   |   |   |
|   |   |   |
|   X   |   |
|   |   |   |
|   |   |   |
(x, y) 좌표: 2 2
---|---|---
|   |   |   |
|   |   |   |
|   X   |   |
|   |   |   |
|   |   |   |
|   |   |   0
---|---|---
(x, y) 좌표: .
```

```
C:\Windows\system32\cmd.exe
산수 문제를 자동으로 출제합니다.
41+67=108
맞았습니다.
계속하려면 아무 키나 누르십시오 . . .
```

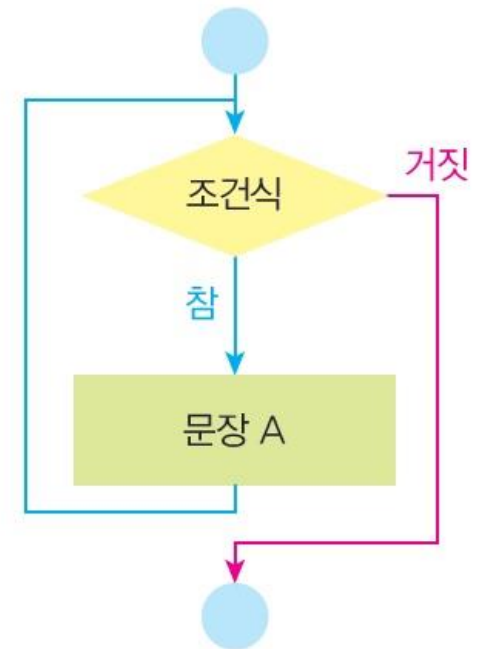
제어구조



순차 구조



선택 구조



반복 구조

표 2.1 관계 연산자

연산자	의미
$x == y$	x와 y가 같은가?
$x != y$	x와 y가 다른가?
$x > y$	x가 y보다 큰가?
$x < y$	x가 y보다 작은가?
$x >= y$	x가 y보다 크거나 같은가?
$x <= y$	x가 y보다 작거나 같은가?

예제

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

int main() {
    bool b;
    b = (1 == 2);

    cout << std::boolalpha;
    cout << b << endl;

    return 0;
}
```



C:\Windows\system32\cmd.exe

```
false
계속하려면 아무 키나 누르십시오 . . .
```

표 2.2 논리 연산자

연산자	의미
$x \ \&\& \ y$	AND 연산, x 와 y 가 모두 참이면 참, 그렇지 않으면 거짓
$x \ \ y$	OR 연산, x 나 y 중에서 하나만 참이면 참, 모두 거짓이면 거짓
$!x$	NOT 연산, x 가 참이면 거짓, x 가 거짓이면 참

if-else문

문법 2.1

if-else 문

```
if ( 조건식 ) {  
    문장1  
}  
else {  
    문장2  
}
```

- 하나의 예로 사용자로 부터 받은 두 개의 정수 중에서 더 큰 수를 찾는 프로그램을 작성하여 보자.



```
C:\Windows\system32\cmd.exe
x값을 입력하시오: 10
y값을 입력하시오: 20
y가 x보다 큽니다.
계속하려면 아무 키나 누르십시오 . . .
```



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

int main() {
    int x, y;

    cout << "x값을 입력하시오: "; // 입력 안내 출력
    cin >> x;

    cout << "y값을 입력하시오: "; // 입력 안내 출력
    cin >> y;

    if (x > y)
        cout << "x가 y보다 큼니다. " << endl;
    else
        cout << "y가 x보다 큼니다. " << endl;
    return 0;
}
```

중첩 if-else 문

문법 2.2

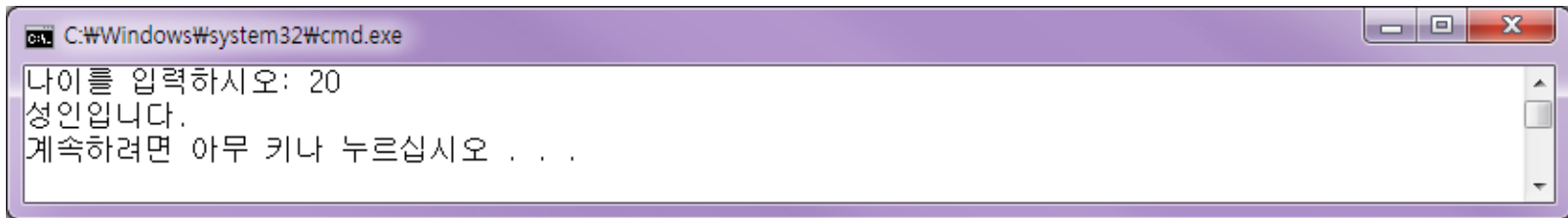
중첩 if-else 문

```
if ( 조건식 ) {  
    문장1  
}  
else if {  
    문장2  
}  
else {  
    문장3  
}
```

- 변수 x 의 값이 양수, 음수, 0인지를 검사하는 코드는 다음과 같다.

```
if (x > 0)
    cout << "x는 양수입니다." << endl;
else if (x < 0)
    cout << "x는 음수입니다." << endl;
else
    cout << "x는 0입니다." << endl;
```

- 사용자로부터 나이를 받아서 어린이, 청소년, 성인을 구분하는 프로그램을 작성하여 보자.



```
C:\Windows\system32\cmd.exe
나이를 입력하시오: 20
성인입니다.
계속하려면 아무 키나 누르십시오 . . .
```

예제

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

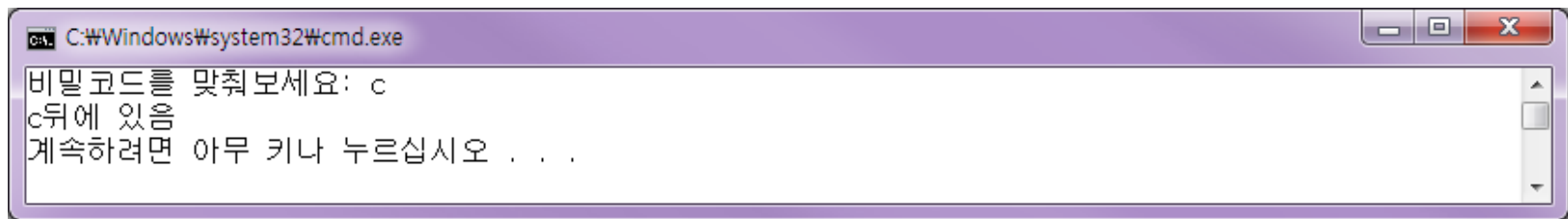
int main()
{
    int age; // 나이
    cout << "나이를 입력하시오: ";
    cin >> age;

    if (age <= 12)
        cout << "어린이입니다." << endl;
    else if (age <= 19)
        cout << "청소년입니다." << endl;
    else
        cout << "성인입니다." << endl;

    return 0;
}
```

Lab: 비밀 코드 맞추기

- 컴퓨터가 숨기고 있는 비밀 코드를 추측하는 게임을 작성해보자. 비밀 코드는 **a**부터 **z** 사이의 문자이다.



Solution

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

int main()
{
    char secret_code = 'h';

    cout << "비밀코드를 맞춰보세요: " ;
    char code;
    cin >> code;
    if (code < secret_code)
        cout << code << "뒤에 있음" << endl;
    else if (code > secret_code)
        cout << code << "앞에 있음" << endl;
    else
        cout << "맞추었습니다." << endl;
    return 0;
}
```

Lab: 세 개의 정수 중에서 큰 수 찾기

- 사용자로부터 받은 3개의 정수 중에서 가장 큰 수를 찾는 프로그램을 작성해보자.



```
C:\Windows\system32\cmd.exe
3개의 정수를 입력하시오: 20 10 30
가장 큰 정수는30
계속하려면 아무 키나 누르십시오 . . .
```


Solution

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

int main()
{
    int a, b, c, largest;

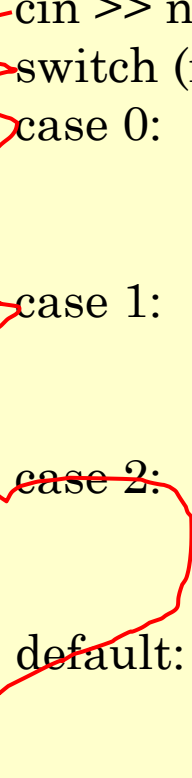
    cout << "3개의 정수를 입력하시오: ";
    cin >> a >> b >> c;

    if (a > b && a > c)
        largest = a;
    else if (b > a && b > c)
        largest = b;
    else
        largest = c;

    cout << "가장 큰 정수는" << largest << endl;
    return 0;
}
```

switch

```
int number;  
cout << "숫자를 입력하시오:";  
cin >> number;  
switch (number) {  
case 0:  
    cout << "zero\n";  
    break;  
case 1:  
    cout << "one\n";  
    break;  
case 2:  
    cout << "two\n";  
    break;  
default:  
    cout << "many\n";  
    break;  
}
```

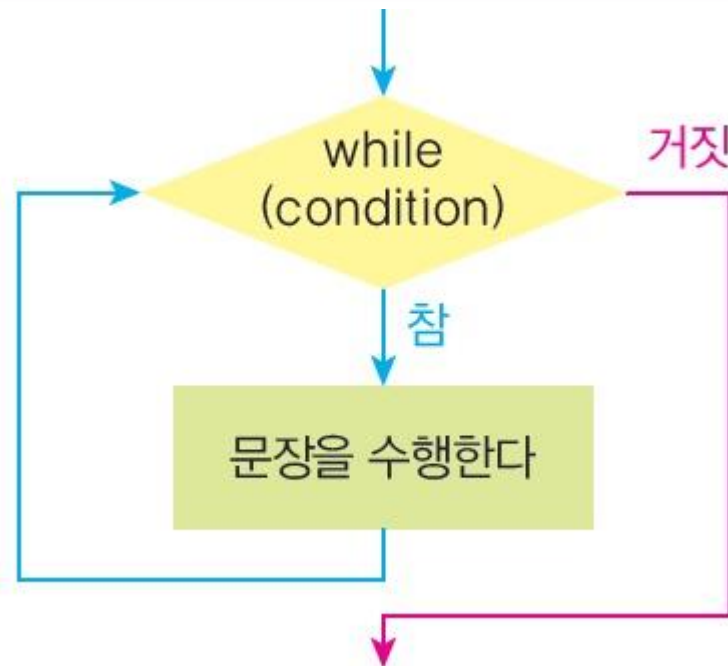


while 루프

문법 2.3

while 루프

```
while (조건식) {  
    문장  
}
```



- 카운트다운 예제를 만들어보자.



```
C:\Windows\system32\cmd.exe
10 9 8 7 6 5 4 3 2 1 발사!
계속하려면 아무 키나 누르십시오 . . .
```

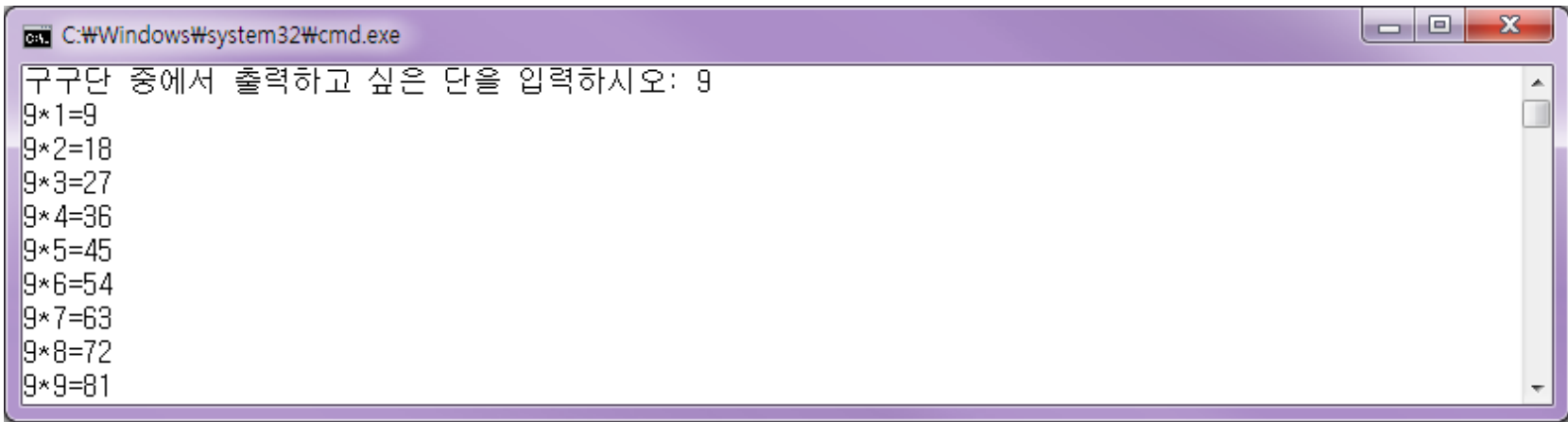
예제

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

int main()
{
    int n = 10;
    while (n > 0) {
        cout << n << " ";
        n--;
    }
    cout << "발사! \n";
}
```

구구단 예제

- 반복문을 사용하여 구구단을 출력하여 보자.



```
C:\Windows\system32\cmd.exe
구구단 중에서 출력하고 싶은 단을 입력하시오: 9
9*1=9
9*2=18
9*3=27
9*4=36
9*5=45
9*6=54
9*7=63
9*8=72
9*9=81
```

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

int main()
{
    int n;
    int i = 1;

    cout << "구구단 중에서 출력하고 싶은 단을 입력하시오: ";
    cin >> n;
    while (i <= 9) {
        cout << n << "*" << i << "=" << n * i << endl;
        i++;
    }
    return 0;
}
```

do-while 루프

문법 2.4

do-while 문

```
do {  
    문장  
} while( 조건식 );
```


- 사용자가 “종료”를 입력하기 전까지는 사용자의 입력을 그대로 화면에 출력하는 프로그램을 작성해 보자.



```
C:\Windows\system32\cmd.exe
문자열을 입력하시오: dog
사용자의 입력: dog
문자열을 입력하시오: cat
사용자의 입력: cat
문자열을 입력하시오: 종료
사용자의 입력: 종료
계속하려면 아무 키나 누르십시오 . . .
```

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

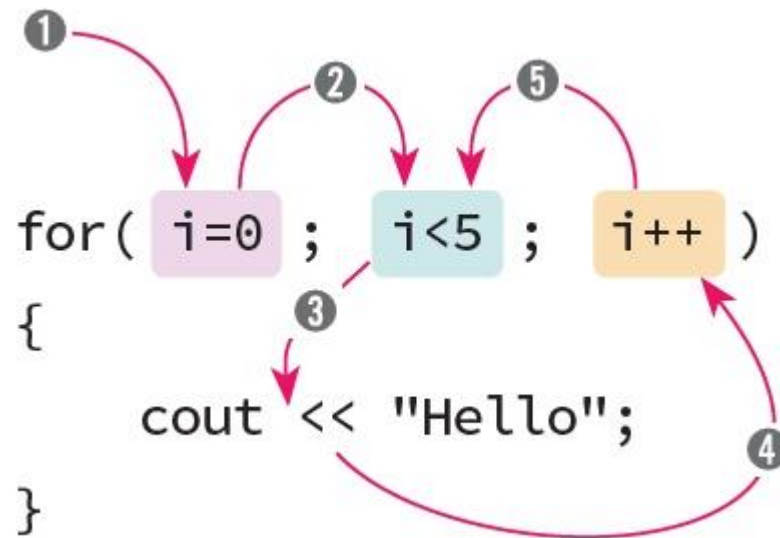
int main()
{
    string str;
    do {
        cout << "문자열을 입력하시오: ";
        getline(cin, str);
        cout << "사용자의 입력: " << str << '\n';
    } while (str != "종료");
    return 0;
}
```

for 루프

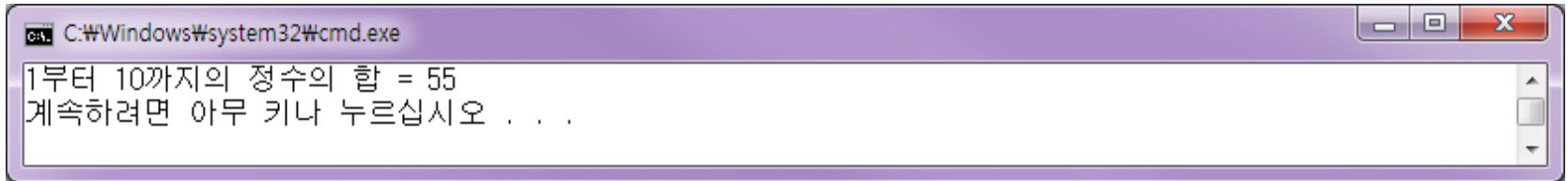
문법 2.5

for 루프

```
for( 초기식 ; 조건식 ; 증감식 ) {  
    문장  
}
```



- 1부터 10까지의 정수를 더하여 합을 구하는 프로그램을 살펴보자.



```
C:\Windows\system32\cmd.exe
1부터 10까지의 정수의 합 = 55
계속하려면 아무 키나 누르십시오 . . .
```

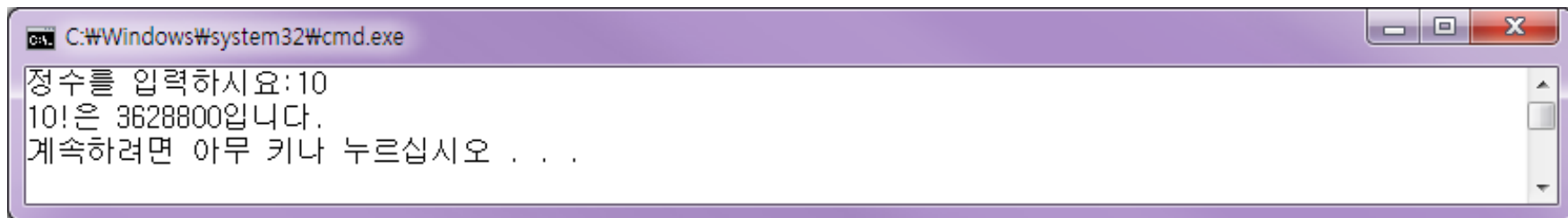
예제

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

int main()
{
    int sum = 0;
    for (int i = 1; i <= 10; i++)
        sum += i;

    cout << "1부터 10까지의 정수의 합 = " << sum << endl;
    return 0;
}
```

- 팩토리얼 값을 계산하여 보자.



```
C:\Windows\system32\cmd.exe
정수를 입력하시요: 10
10!은 3628800입니다.
계속하려면 아무 키나 누르십시오 . . .
```

예제

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

int main()
{
    long fact = 1;
    int n;

    cout << "정수를 입력하세요:";
    cin >> n;

    for (int i = 1; i <= n; i++)
        fact = fact * i;

    cout << n << "!은 " << fact << "입니다.\n";

    return 0;
}
```

break 문장 예제

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

int main()
{
    for (int i = 1; i < 10; i++)
    {
        cout << i << " ";
        if (i == 4)
            break;
    }
    return 0;
}
```



continue 문장 예제

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

int main()
{
    int i = 0;
    do {
        i++;
        cout << "continue 문장 전에 있는 문장" << endl;
        continue;
        cout << "continue 문장 후에 있는 문장" << endl;
    } while (i < 3);
    return 0;
}
```

C:\Windows\system32\cmd.exe

```
continue 문장 전에 있는 문장
continue 문장 전에 있는 문장
continue 문장 전에 있는 문장
계속하려면 아무 키나 누르십시오 . . .
```

Lab: 자음과 모음 개수 세기

- 사용자로부터 영문자를 받아서 자음과 모음의 개수를 세는 프로그램을 작성하여 보자.



```
C:\Windows\system32\cmd.exe
영문자를 입력하고 콘트롤투Z를 치세요
a
b
c
d
e
^Z
모음: 2
자음: 3
```

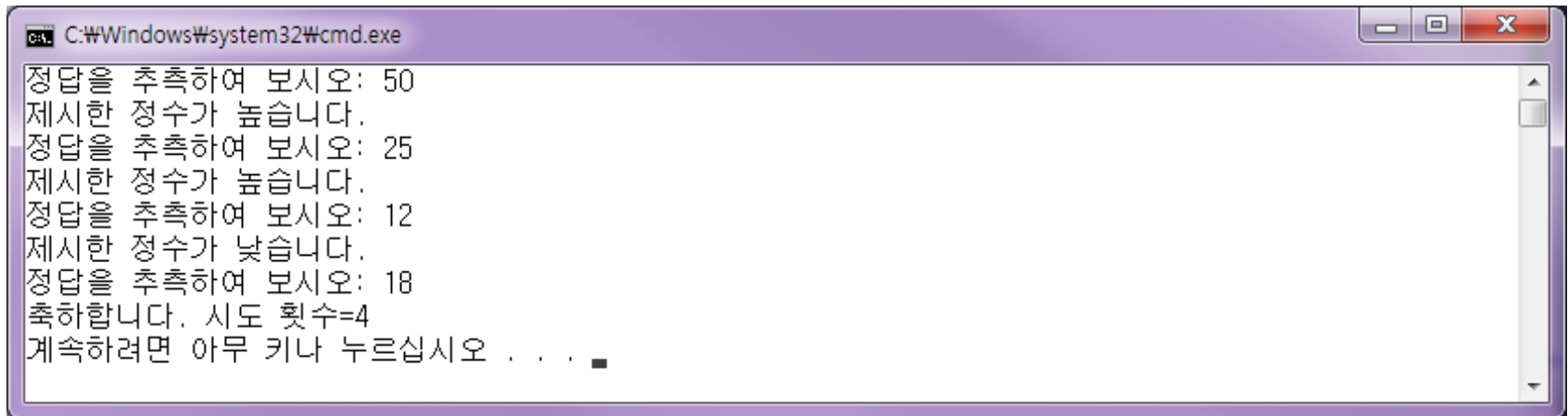
```
#include <iostream>
using namespace std;

int main()
{
    int vowel = 0, consonant = 0;
    char ch;

    cout << "영문자를 입력하고 콘트롤-Z를 치세요" << endl;
    while (cin >> ch) {
        switch (ch) {
            case 'a': case 'i': case 'e': case 'o': case 'u':
                vowel++;          break;
            default:
                consonant++;      break;
        }
    }
    cout << "모음: " << vowel << endl;
    cout << "자음: " << consonant << endl;
    return 0;
}
```

Lab: 숫자 맞추기 게임

- 프로그램은 1부터 100사이의 정수를 저장하고 있고 사용자는 질문을 통하여 그 정수를 알아맞히려고 노력한다.



```
C:\Windows\system32\cmd.exe
정답을 추측하여 보시오: 50
제시한 정수가 높습니다.
정답을 추측하여 보시오: 25
제시한 정수가 높습니다.
정답을 추측하여 보시오: 12
제시한 정수가 낮습니다.
정답을 추측하여 보시오: 18
축하합니다. 시도 횟수=4
계속하려면 아무 키나 누르십시오 . . .
```

Solution

```
// 반복을 이용한 게임 프로그램
#include <iostream>
#include <time.h>
using namespace std;

int main()
{
    srand(time(NULL));

    int answer = rand() %100;           // 정답
    int guess;
    int tries = 0;
    // 반복 구조
```

Solution

```
do {  
    cout << "정답을 추측하여 보시오: ";  
    cin >> guess;  
    tries++;  
  
    if (guess > answer)  
        cout << "제시한 정수가 높습니다.\n";  
    if (guess < answer)  
        cout << "제시한 정수가 낮습니다.\n";  
} while (guess != answer);  
  
cout << "축하합니다. 시도 횟수=" << tries << endl;  
  
return 0;  
}
```

Lab: 산수 문제 자동 출제

- 초등학교 학생들을 위한 산수 문제를 자동으로 출제하는 프로그램을 작성해보자.



```
C:\Windows\system32\cmd.exe
산수 문제를 자동으로 출제합니다.
41+67=108
맞았습니다.
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <iostream>
#include <stdlib.h>
using namespace std;

int main()
{
    int i, ans;
    cout << "산수 문제를 자동으로 출제합니다." << endl;

    while (true) {
        int x = rand() % 100;
        int y = rand() % 100;
        cout << x << "+" << y << "=";
        cin >> ans;
        if (x + y == ans) {
            cout << "맞았습니다." << endl;
            break;
        }
        else
            cout << "틀렸습니다." << endl;
    }
    return 0;
}
```


- 배열(array)은 같은 종류의 데이터들이 순차적으로 메모리에 저장되는 자료 구조이다

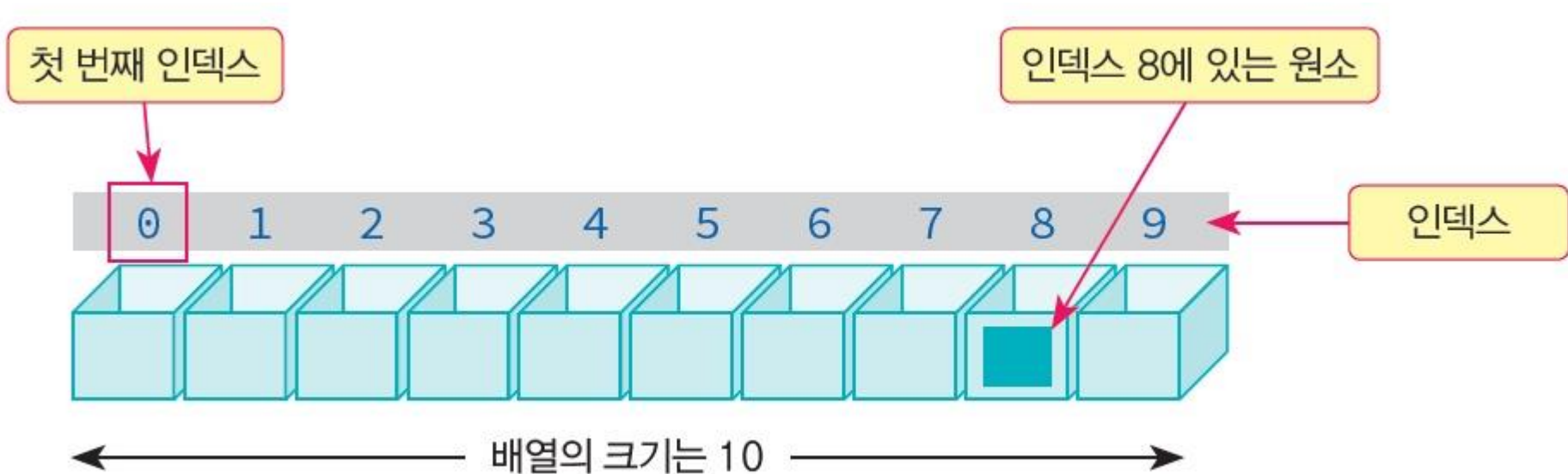


그림 2.1 배열의 개념

문법 2.6

배열 선언



The diagram illustrates the syntax for declaring an array in C. It shows the code `int scores[10];` with annotations: a yellow box labeled "배열의 이름" (Array Name) points to `scores`, and another yellow box labeled "배열의 크기" (Array Size) points to `10`. The `int` keyword is highlighted in light blue, and the closing semicolon is highlighted in light orange.

```
int scores[10];
```

예제

- 학생들의 성적을 입력받아서 평균을 구하는 프로그램을 배열을 이용하여서 작성하여 보자.



```
C:\Windows\system32\cmd.exe
학생들의 성적을 입력하시오: 10
학생들의 성적을 입력하시오: 20
학생들의 성적을 입력하시오: 30
학생들의 성적을 입력하시오: 40
학생들의 성적을 입력하시오: 50
학생들의 성적을 입력하시오: 60
학생들의 성적을 입력하시오: 70
학생들의 성적을 입력하시오: 80
학생들의 성적을 입력하시오: 90
학생들의 성적을 입력하시오: 100
성적 평균= 55
```

예제

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

int main() {
    const int STUDENTS = 10;
    int scores[STUDENTS];
    int sum = 0;
    int i, average;

    for (i = 0; i < STUDENTS; i++) {
        cout << "학생들의 성적을 입력하시오: ";
        cin >> scores[i];
    }
    for (i = 0; i < STUDENTS; i++)
        sum += scores[i];

    average = sum / STUDENTS;
    cout << "성적 평균= " << average << endl;
    return 0;
}
```

배열의 초기화

```
int sales[5] = { 100, 200, 300, 400, 500 };  
int sales[5] = { 100, 200, 300 };  
int sales[] = { 100, 200, 300, 400, 500, 600, 700 };
```

	0	1	2	3	4
sales	100	200	300	400	500

- 모든 초기화에 중괄호 { ... }을 사용한다.

```
int scores[] { 10, 20, 30 };
```

```
int a{ 0 }; // int a=0;과 동일하다.
```

```
string s{ "hello" }; // string s="hello";
```

```
vector<string> list{ "alpha", "beta", "gamma" }; // 벡터 생성시 초기화
```

- 이 형식은 배열 안에 들어 있는 모든 값에 대하여 반복할 때 사용된다.

문법 2.7

범위-기반 for 루프

```
for( 변수 : 범위 ) {  
    문장  
}
```

예제

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

int main()
{
    int list[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    for (int i : list) {
        cout << i << " ";
    }
    cout << endl;
}
```



```
for (int& i : list) {  
    cout << i << " ";  
}
```

```
for (auto& i : list) {  
    cout << i << " ";  
}
```

예제

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

int main()
{
    int list[] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    for (int& i : list) {
        i = i*i;
    }
    for (int i : list) {
        cout << i << " ";
    }
}
```

배열 요소의 변경이
필요하다면 반드시
참조자를 사용하여야 한다.

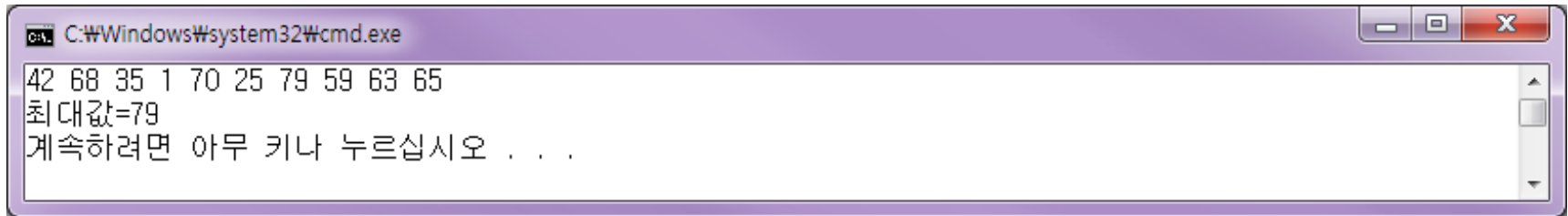
C:\Windows\system32\cmd.exe

1 4 9 16 25 36 49 64 81 100

계속하려면 아무 키나 누르십시오 . . .

Lab: 배열에서 최대값 찾기

- 크기가 100인 배열을 1부터 100 사이의 난수로 채우고
배열 요소 중에서 최대값을 찾아보자



```
C:\Windows\system32\cmd.exe
42 68 35 1 70 25 79 59 63 65
최대값=79
계속하려면 아무 키나 누르십시오 . . .
```

Solution

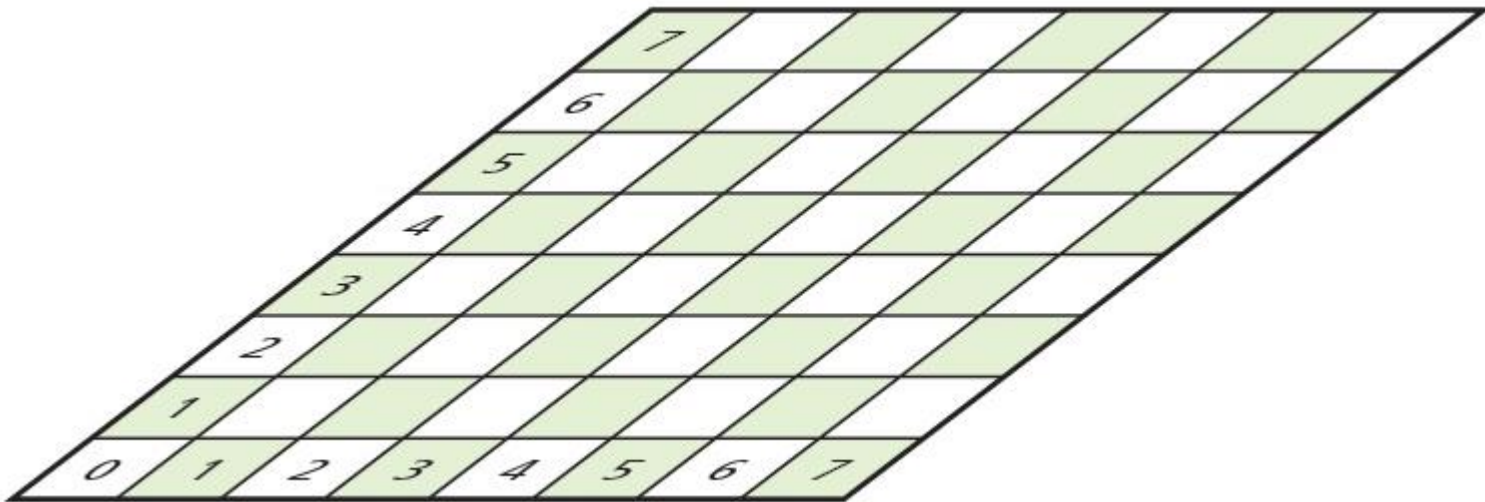
```
#include <iostream>
#include <stdlib.h>
using namespace std;

int main() {
    int list[10];
    int max;

    for (int& elem : list) {
        elem = rand() % 100 + 1;
        cout << elem << " ";
    }
    cout << endl;
    max = list[0];
    for (auto& elem : list) {
        if (elem > max)
            max = elem;
    }
    cout << "최대값=" << max << endl;
    return 0;
}
```

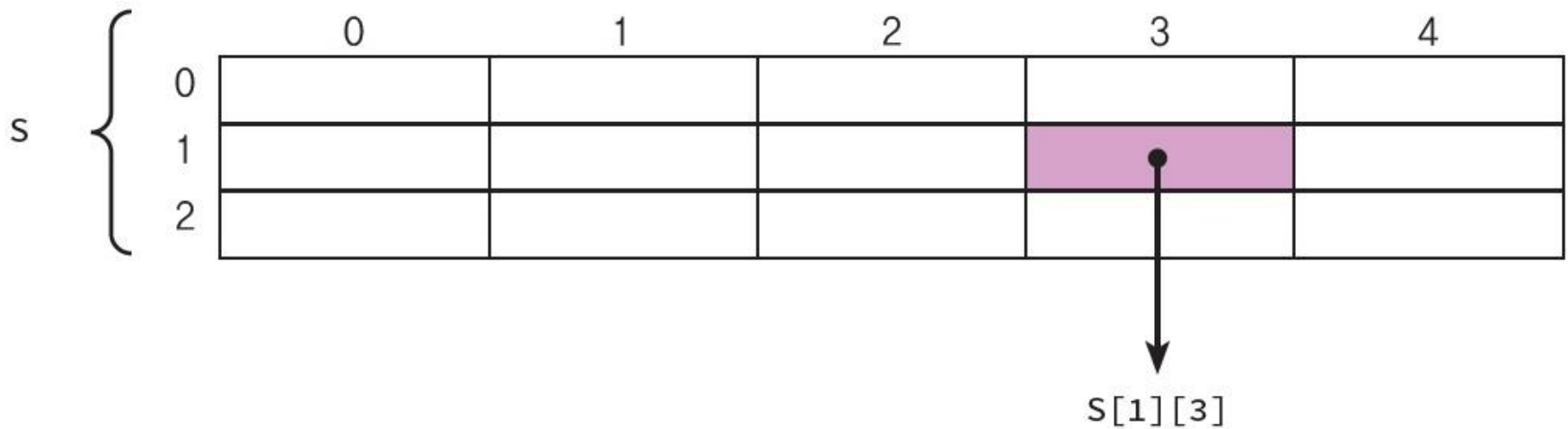
2차원 배열

- 배열 요소들이 2차원으로 나열된 배열



2차원 배열의 선언

```
int s[3][5];
```



2차원 배열의 초기화

```
int s[3][5] = {  
    { 1, 2, 3, 4, 5 },      // 첫 번째 행의 요소들의 초기값  
    { 2, 4, 6, 8, 10 },    // 두 번째 행의 요소들의 초기값  
    { 3, 6, 9, 12, 15 }    // 세 번째 행의 요소들의 초기값  
};
```

		0	1	2	3	4
s	0	1	2	3	4	5
	1	2	4	6	8	10
	2	3	6	9	12	15

□ 구구단의 저장



```
C:\Windows\system32\cmd.exe  
1, 2, 3, 4, 5, 6, 7, 8, 9,  
2, 4, 6, 8, 10, 12, 14, 16, 18,  
3, 6, 9, 12, 15, 18, 21, 24, 27,  
계속하려면 아무 키나 누르십시오 . . .
```


예제

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

#define WIDTH 9
#define HEIGHT 3

int main() {
    int table[HEIGHT][WIDTH];
    int r, c;

    for (r = 0; r < HEIGHT; r++)
        for (c = 0; c < WIDTH; c++)
            table[r][c] = (r + 1)*(c + 1);

    for (r = 0; r < HEIGHT; r++) {
        for (c = 0; c < WIDTH; c++) {
            cout << table[r][c] << ", ";
        }
        cout << endl;
    }
}
```

Lab: Tic-Tac-Toe 게임

- Tic-Tac-Toe 게임은 2명의 경기자가 오른쪽과 같은 보드를 이용하여서 번갈아가며 O와 X를 놓는 게임이다.



```
C:\Windows\system32\cmd.exe
(x, y) 좌표: 1 1
---|---|---
|   |   |   |
|   |   |   |
| X |   |   |
|   |   |   |
|   |   |   |
(x, y) 좌표: 2 2
---|---|---
|   |   |   |
|   |   |   |
| X |   |   |
|   |   |   |
|   |   |   |
|   |   | 0 |
(x, y) 좌표: .
```

예제

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

int main()
{
    char board[3][3];
    int x, y, k, i;

    // 보드를 초기화한다.
    for (x = 0; x < 3; x++)
        for (y = 0; y < 3; y++) board[x][y] = ' ';

    // 사용자로부터 위치를 받아서 보드에 표시한다.
    for (k = 0; k < 9; k++) {
        cout << "(x, y) 좌표: ";
        cin >> x >> y;
        board[x][y] = (k % 2 == 0) ? 'X' : 'O';
    }
}
```

```
// 보드를 화면에 그린다.  
for (i = 0; i < 3; i++) {  
    cout << "--- | --- | ---" << endl;  
    cout << board[i][0] << " | " << board[i][1] << " | "  
    << board[i][2] << endl;  
}  
    cout << "--- | --- | ---" << endl;  
}  
  
return 0;  
}
```