

민선생코딩학원 시작반

수업노트 LV-06



배우는 내용

다중 반복문의 시작

1. char 응용
2. 2중 for문 시작하기

ASCII코드란?

- ▶ 컴퓨터는 물리적으로 글자를 저장할 수 없기에 숫자로 바꾸어서 저장한다.
- ▶ 컴퓨터에 문자를 저장하기 위해, 아스키(ASCII) 라는 협회에서 문자를 숫자로 나타내는 표를 만들었고, 이는 전세계에서 공용적으로 쓰인다.
- ▶ ASCII코드(아스키코드)는 기본적으로 128개로 구성되어 있다.
ex) 대문자 A는 아스키코드 65번
ex) 문자 '0'은 아스키코드 48번
ex) 소문자 a는 아스키코드 97번

| ASCII TABLE | | | | | | | | | | | |
|-------------|-----|------------------------|---------|-----|---------|---------|-----|------|---------|-----|-------|
| Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char | Decimal | Hex | Char |
| 0 | 0 | [NULL] | 32 | 20 | [SPACE] | 64 | 40 | @ | 96 | 60 | ` |
| 1 | 1 | [START OF HEADING] | 33 | 21 | ! | 65 | 41 | A | 97 | 61 | a |
| 2 | 2 | [START OF TEXT] | 34 | 22 | " | 66 | 42 | B | 98 | 62 | b |
| 3 | 3 | [END OF TEXT] | 35 | 23 | # | 67 | 43 | C | 99 | 63 | c |
| 4 | 4 | [END OF TRANSMISSION] | 36 | 24 | \$ | 68 | 44 | D | 100 | 64 | d |
| 5 | 5 | [ENQUIRY] | 37 | 25 | % | 69 | 45 | E | 101 | 65 | e |
| 6 | 6 | [ACKNOWLEDGE] | 38 | 26 | & | 70 | 46 | F | 102 | 66 | f |
| 7 | 7 | [BELL] | 39 | 27 | ' | 71 | 47 | G | 103 | 67 | g |
| 8 | 8 | [BACKSPACE] | 40 | 28 | (| 72 | 48 | H | 104 | 68 | h |
| 9 | 9 | [HORIZONTAL TAB] | 41 | 29 |) | 73 | 49 | I | 105 | 69 | i |
| 10 | A | [LINE FEED] | 42 | 2A | * | 74 | 4A | J | 106 | 6A | j |
| 11 | B | [VERTICAL TAB] | 43 | 2B | + | 75 | 4B | K | 107 | 6B | k |
| 12 | C | [FORM FEED] | 44 | 2C | , | 76 | 4C | L | 108 | 6C | l |
| 13 | D | [CARRIAGE RETURN] | 45 | 2D | - | 77 | 4D | M | 109 | 6D | m |
| 14 | E | [SHIFT OUT] | 46 | 2E | . | 78 | 4E | N | 110 | 6E | n |
| 15 | F | [SHIFT IN] | 47 | 2F | / | 79 | 4F | O | 111 | 6F | o |
| 16 | 10 | [DATA LINK ESCAPE] | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p |
| 17 | 11 | [DEVICE CONTROL 1] | 49 | 31 | 1 | 81 | 51 | Q | 113 | 71 | q |
| 18 | 12 | [DEVICE CONTROL 2] | 50 | 32 | 2 | 82 | 52 | R | 114 | 72 | r |
| 19 | 13 | [DEVICE CONTROL 3] | 51 | 33 | 3 | 83 | 53 | S | 115 | 73 | s |
| 20 | 14 | [DEVICE CONTROL 4] | 52 | 34 | 4 | 84 | 54 | T | 116 | 74 | t |
| 21 | 15 | [NEGATIVE ACKNOWLEDGE] | 53 | 35 | 5 | 85 | 55 | U | 117 | 75 | u |
| 22 | 16 | [SYNCHRONOUS IDLE] | 54 | 36 | 6 | 86 | 56 | V | 118 | 76 | v |
| 23 | 17 | [ENG OF TRANS. BLOCK] | 55 | 37 | 7 | 87 | 57 | W | 119 | 77 | w |
| 24 | 18 | [CANCEL] | 56 | 38 | 8 | 88 | 58 | X | 120 | 78 | x |
| 25 | 19 | [END OF MEDIUM] | 57 | 39 | 9 | 89 | 59 | Y | 121 | 79 | y |
| 26 | 1A | [SUBSTITUTE] | 58 | 3A | : | 90 | 5A | Z | 122 | 7A | z |
| 27 | 1B | [ESCAPE] | 59 | 3B | ; | 91 | 5B | [| 123 | 7B | { |
| 28 | 1C | [FILE SEPARATOR] | 60 | 3C | < | 92 | 5C | \ | 124 | 7C | |
| 29 | 1D | [GROUP SEPARATOR] | 61 | 3D | = | 93 | 5D |] | 125 | 7D | } |
| 30 | 1E | [RECORD SEPARATOR] | 62 | 3E | > | 94 | 5E | ^ | 126 | 7E | ~ |
| 31 | 1F | [UNIT SEPARATOR] | 63 | 3F | ? | 95 | 5F | _ | 127 | 7F | [DEL] |


문자 '0'과 숫자 0은 다르다.
'0'은 ASCII코드 번호 (48번)를 갖고있는 문자이다.

Character 변수의 특징

```
char min = 'A';
```

min 변수에 'A'가 저장되는 것 처럼 보이지만,
실제로는 **아스키코드 값 65**가 저장 된다.

int 타입과 char 타입 모두 숫자를 저장하는 변수이다.
하지만, **가독성을 위해** 목적에 맞게 구분해서 쓰자.

| 조사식 1 | |
|---|--------|
| 이름 | 값 |
|  min | 65 'A' |

```
char b = 65;  
cout << b;
```

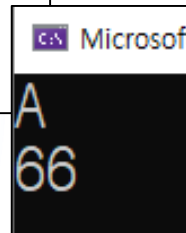
실행결과 : A

ASCII코드 표 없이 ASCII코드 값 알아내기

방법1

아래와 같이 출력할 때 (int)(변수) 로 쓴다.
val2를 int형으로 취급하도록 하는 방법이다.

```
char val1 = 'A';  
cout << val1 << endl;  
  
char val2 = 'B';  
cout << (int)(val2) << endl;
```



A
66

방법2

디버깅 모드 진입 후,
조사식에 'A' 를 등록하면 ASCII코드 값을 확인할 수 있다.

| 조사식 1 | |
|-------|--------|
| 이름 | 값 |
| 'A' | 65 'A' |
| 'B' | 66 'B' |
| 'C' | 67 'C' |

문자 '0' ~ '9' 의 ASCII코드 값

ASCII코드에는 한자리 숫자를 표현하는
문자 '0' ~ '9'도 존재 한다.

조사식 1

| 이름 | 값 |
|-----|--------|
| '0' | 48 '0' |
| '1' | 49 '1' |
| '2' | 50 '2' |

숫자 0과 문자 '0'은 서로 다르다.

```
char ch1 = 48;  
char ch2 = '0';
```

ch1과 ch2는 같은 값이다.

```
char vh1 = '0';  
int vh2 = 0;
```

vh1과 vh2는 다른 값이다.

대문자 or 한자리 숫자인지 구분하기

문자 하나를 입력 받자.

그리고 입력 받은 문자가 **대문자인지 or 한자리 숫자인지** 구분해보자.

```
char ch;  
cin >> ch;  
  
if (ch >= 'A' && ch <= 'Z') {  
    // 대문자  
}  
  
if (ch >= '0' && ch <= '9') {  
    // 한자리 숫자  
}
```

'0' ~ '9' 사이에 있는 값이면
숫자형태로 된 문자이다.

A부터 Z까지 출력하기

방법1

문자 A의 ASCII 값은 65번, 문자 Z의 ASCII 값은 90번
이기에 아래와 같이 코딩이 가능하다.

```
int x;

for (x=65; x<=90; x++)
{
    cout << (char)(x);
}
```

int 타입 변수를 char로 인식하게 만든다. (이를 캐스팅이라고 한다.)
만약 cout << x; 로 썼다면, 65 ~ 90 숫자가 출력 될 것이다.

방법2

'A' 를 소스코드에 적으면, 이를 65로 받아들인다.
따라서 아래와 같은 코딩이 가능하며
65대신 'A'를 적는 것이 가독성에 더 좋다.

```
char x;

for (x='A'; x<='Z'; x++)
{
    cout << x;
}
```

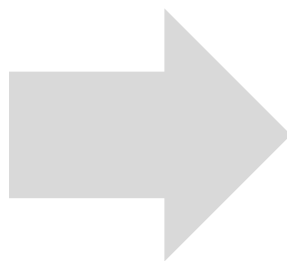

char 변수 사용시 자주하는 실수

char 변수 x를 문자 a로 초기화 후 출력하자.

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

int main()
{
    char x = a;
    cout << x;

    return 0;
}
```



a라는 변수를 찾을 수 없기 때문에
Compile Error가 발생한다.

문자 나타낼 때는 따옴표를 붙여주자.
char x = 'a'로 수정해야 한다.

대문자를 소문자로 바꾸는 방법

대문자에 32를 더하면 소문자로 변환 가능하다.

'A'는 65번이고 소문자 'a'는 97번의 아스키 값을 가진다.

'B'는 66번이고 소문자 'b'는 98번의 아스키 값을 가진다.

```
char ch;  
  
cin >> ch;  
cout << (char)(ch + 32);
```

아스키코드 번호 없이 대문자를 소문자로 변환하는 방법

```
char ch;  
  
cin >> ch;  
cout << (char)(ch - 'A' + 'a');
```

만약 'B' 를
입력 받았다면?

```
'B' - 'A' + 'a'  
= 1 + 'a'  
= 'b'
```

'문자'로 된 '숫자' 구분 하기

문자로 입력 받은 숫자는 '0' ~ '9' 로 쓴다.

```
if (input >= '0' && input <= '9')
{
    // 숫자
}
```

문자를 하나 입력 받고 아래와 같이 처리하자.

- 1) 소문자이면 대문자로 변경해서 출력
- 2) 대문자이면 소문자로 변경해서 출력
- 3) 숫자 형태 문자이면 15를 더한 값 출력

```
char ch;

if (ch >= 'a' && ch <= 'z')
{
    cout << (ch - 'a') + 'A';
}
if (ch >= 'A' && ch <= 'Z')
{
    cout << (ch - 'A') + 'a';
}
if (ch >= '0' && ch <= '9')
{
    cout << (ch - '0') + 15;
}
```

char 배열 응용하기 1

문자 H부터 C까지 배열에 문자를 순차적으로 채우고 출력하기

| | | | | | | | |
|---|---|---|---|---|---|--|--|
| H | G | F | E | D | C | | |
|---|---|---|---|---|---|--|--|

```
char vect[8];
```

```
int t = 0;  
for (char x = 'H'; x >= 'C'; x--) {  
    vect[t] = x;  
    t++;  
}
```

```
for (int i = 0; i < t; i++) {  
    cout << vect[i] << " ";  
}
```

2중 for문 시작하기

2중 for문이란 1중 for문 소스코드를 n번 반복시키는 코드이다.
2중 for문을 이용하여 1~3을 반복적으로 출력하자.

```
for (x=1; x<=3; x++)  
{  
    cout << x;  
}
```

출력결과 : 123

4회 반복

```
for (y=0; y<4; y++)  
{  
    for (x=1; x<=3; x++)  
    {  
        cout << x;  
    }  
}
```

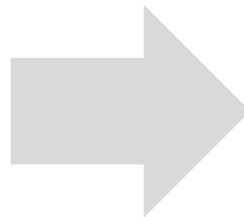
출력결과 : 123123123123

2중 for문 사용 예제

2중 for문을 사용하여 #을 3x4로 출력하자.

```
for(3회 반복)
{
    for (x = 0; x<4; x++)
    {
        cout << "#";
    }
    cout << endl;
}
```

#을 4회 출력하는 소스코드



실행결과

```
####
####
####
```

2중 for문 출력 예제

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

int main()
{
    int y;
    int x;

    for (y = 0; y < 3; y++)
    {
        for (x = 0; x < 4; x++)
        {
            cout << "#";
        }
        cout << endl;
    }

    return 0;
}
```

출력결과 :

####

```
int x, y;
int t = 1;

for (y = 0; y < 2; y++) {
    for (x = 0; x < 3; x++) {
        cout << t << " ";
        t++;
    }
    cout << endl;
}
```

출력결과 :

1 2 3
4 5 6

2중 for문 응용하기 1

1부터 4까지, '1234' 를 세줄 출력하는 소스코드를 작성하자.

```
for (x=1; x<=4; x++)  
{  
    cout << x;  
}  
cout << endl;
```

3회 반복

```
for (y=0; y<3; y++)  
{  
    for (x=1; x<=4; x++)  
    {  
        cout << x;  
    }  
    cout << endl;  
}
```

출력결과 :
1234
1234
1234

2중 for문 돌릴 때는 변수 2개를 사용하는 것을 권장한다.

2중 for문 응용하기 2

한 줄마다 1부터 4까지 하나의 숫자로 출력하는 소스코드를 작성하자.

```
for (y=1; y<=4; y++)  
{  
    for (x=0; x<5; x++)  
    {  
        cout << y;  
    }  
    cout << endl;  
}
```

```
for (y=0; y<4; y++)  
{  
    for (x=0; x<5; x++)  
    {  
        cout << y + 1;  
    }  
    cout << endl;  
}
```

```
int t = 1;  
for (y=0; y<4; y++)  
{  
    for (x=0; x<5; x++)  
    {  
        cout << t;  
    }  
    t++;  
    cout << endl;  
}
```

출력결과 :

11111
22222
33333
44444

복잡한 2중 for문 소스코드 이해하기

다음 소스코드를 Trace를 통해 어떻게 동작하는지 이해 해보자.

```
for (y=0; y<2; y++)  
{  
    for (x=0; x<=3; x++)  
    {  
        cout << y << " " << x << endl;  
    }  
}
```

출력결과 :

```
0 0  
0 1  
0 2  
0 3  
1 0  
1 1  
1 2  
1 3
```