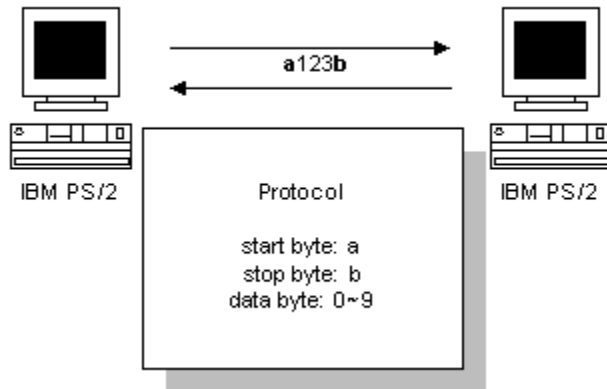




## 5. 이스케이프 절차Escape Sequence

- 통신 규약(프로토콜, Protocol)



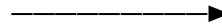
통신규약Protocol: 컴퓨터 사이의 데이터 전송을 위해 두 대의 컴퓨터는 미리 정해진 약속에 따라 서로 통신합니다.

- 
- a를 시작 바이트start byte, b를 끝 바이트stop byte로 사용하기로 하였습니다. 그러므로 123을 전송하려면 아래와 같이 데이터를 전송합니다.

**a123b**

- 왼쪽에서 오른쪽으로 데이터가 전송된다면 아래와 같이 나타내야 할 것입니다.

**b 3 2 1 a**



- 영문자도 데이터에 포함시키도록 요구사항이 추가된 것입니다.

**a12a3b**

- 
- 다음과 같이 프로토콜을 개선하였습니다.

(1) 시작 바이트와 끝 바이트는 각각 'ea', 'eb'로 나타냅니다.

(2) e가 데이터에 포함되는 경우는 'ee'로 나타냅니다.

- 전송해야할 데이터가, 12a3인 경우는 아래와 같이 표현할 수 있습니다.

**ea12a3eb**

- 데이터가 12e3a인 경우는 아래와 같이 전송합니다.

**ea12ee3aeb**

---

- 다음과 같이, 시작 바이트, 끝 바이트 혹은 데이터 자체 e를 구분하는 것이 가능합니다.

- (1) e 다음에 a가 오면 시작 바이트
- (2) e 다음에 b가 오면 끝 바이트
- (3) e 다음에 e가 오면 데이터 e

- C/C++에서 스트링string은 이러한 Esc 절차를 이용하여 표현합니다.
- 어떠한 표현이 Esc 절차를 사용한다고 하면, 그 표현이 사용하는 Esc 문자가 무엇인지를 확인해야 하며, Esc 다음에 사용하는 약속된 문자에는 무엇이 있는지 확인해야 합니다. C++에서는 역슬래시back-slash '\`'를 문자열 표현에서 Esc 문자로 사용합니다.
- printf() 같은 몇 함수는 숫자를 나타내기 위해 '%'도 Esc 문자로 사용합니다.
- Esc 다음에 올 수 있는 약속된 문자에는 다음과 같은 문자가 있습니다.

n, a, b, t, v

- "WaaWWWnn"을 해석해 보면, 아래와 같은 Esc 절차로 분리됩니다.

\`a + a + \`W\`W + \`Wn + n

- 
- "...\\n..."이 항상 10번 문자로 해석될 것을 기대해서도 안 됩니다.

```
printf("hello\\nworld");
```

- 위의 소스는 다음과 같이 출력됩니다.

```
hello\\nworld
```

- 
- 콘솔 화면의 임의의 위치에 문자열을 출력하기 위해서는 SetConsoleCursorPosition()이라는 Win32 Api(Application Programming Interface) 함수를 사용해야 합니다.

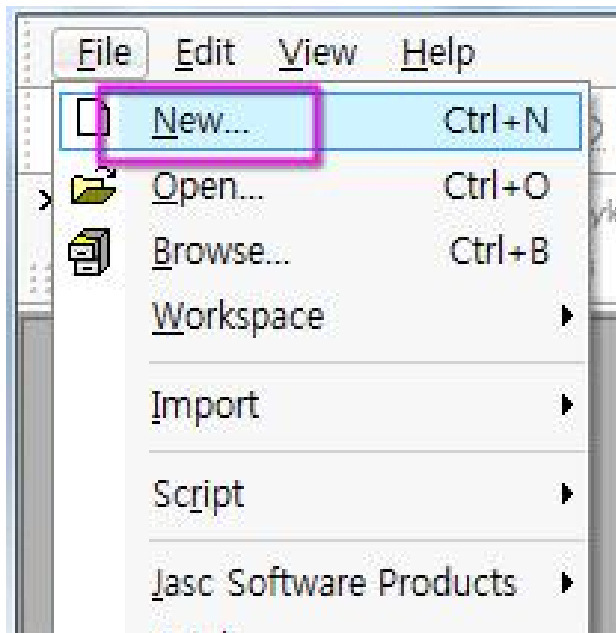
```
void GotoXy( int x, int y )  
{  
    COORD p = { x, y };  
    SetConsoleCursorPosition( GetStdHandle( STD_OUTPUT_HANDLE ), p );  
} //GotoXy()
```



## 특정한 문자를 강조(Highlight)하기

"&New..."

- 메뉴 항목은 N을 접근키로 등록하라는 지시입니다.





- PutHighlightedString()의 소스를 아래에 리스트 하였습니다.

```
#include <stdio.h>
#include <conio.h>
#include <windows.h>

enum EConsoleTextColor
{
    BLACK,
    BLUE,
    GREEN,
    CYAN,
    RED,
    MAGENTA,
    YELLOW,
    WHITE,
}; //enum EConsoleTextColor

void GotoXy( int x, int y )
```

---

```
{
    COORD p = { x, y };
    SetConsoleCursorPosition( GetStdHandle( STD_OUTPUT_HANDLE ), p );
} //GotoXy()

void SetColor( EConsoleTextColor iForegroundColor_, bool bForegroundLighten_
    , EConsoleTextColor iBackgroundColor_, bool bBackgroundLighten_ )
{
    WORD wForeground = iForegroundColor_
    if(bForegroundLighten_ == true)
        wForeground |= 0x8;
    WORD wBackground = iBackgroundColor_ << 4;
    if(bBackgroundLighten_ == true)
        wBackground |= 0x80;
    SetConsoleTextAttribute( GetStdHandle( STD_OUTPUT_HANDLE ), wForeground |
wBackground );
} //SetColor()

void PutHighlightedString( char* pszText_, EConsoleTextColor eHighlightColor_ )
```

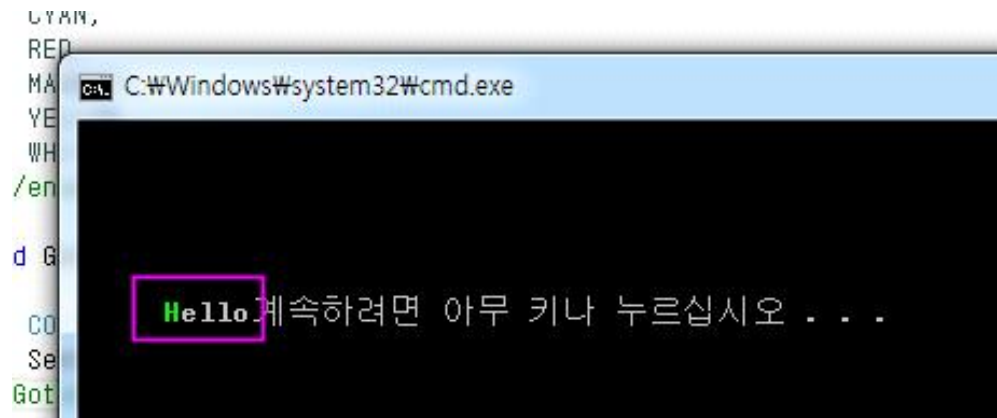
---

```
{
    const EConsoleTextColor eNormalColor = WHITE
    const EConsoleTextColor eBackColor = BLACK
    const int ciStrLength = strlen( pszText_ );
    int iPos = 0;
    while( iPos < ciStrLength )
    {
        char ch = pszText_[ iPos ];
        if( ch == '&' )
        {
            iPos += 1;
            ch = pszText_[ iPos ];
            SetColor( eHighlightColor_, true, eBackColor, false );
        }
        else
        {
            SetColor( eNormalColor, false, eBackColor, false );
        }
    }
    printf( "%c", ch );
}
```

```
        iPos += 1;
    }//while
} //PutHighlightedString()

void main()
{
    //system( "CLS" );
    GotoXy( 5, 5 );
    PutHighlightedString( "&Hello", GREEN );
} //main()
```

- 출력결과는 아래 그림과 같습니다.





## C++11: 원시 문자열raw string

- C++11은 원시 문자열을 지원합니다. 원시 문자열은 다음과 같이 표현해야 합니다.

`R"[id](<문자>[...])[id]"`

- 대문자 R다음에 " 가 위치합니다. 그리고 사용자가 자유롭게 정할 수 있는 id가 위치하거나 생략할 수 있습니다.
- 그리고 ( 를 명시합니다. 여기까지가 원시 문자열의 시작을 나타내는 문자순서입니다. 그리고 문자열을 구성하는 문자들을 자유롭게 구성합니다.
- 마지막에는 ) 를 적고, 반드시 원시 문자열의 시작을 명시할 때 사용한 id를 적어준 다음에 " 를 적습니다.

- 
- 아래의 표현은 모두 같은 원시 문자열 표현입니다.

**R"idhere(Digital \r\n Contents2\r\n)idhere"**

**R"(Digital \r\n Contents2\r\n)"**

**"Digital \r\n Contents2\r\n"**

- 아래의 코드는 간단한 문자열 리터럴을 출력합니다.

```
#include <stdio.h>
#include <iostream>

void main()
{
    char* pText = "Digital \r\n Contents\r\n"
    char* pText2 = R"(Digital \r\n Contents2\r\n)"
    printf("%s\r\n", pText);
    printf("%s\r\n", pText2);
    /*
    Digital
    Contents

    Digital \r\n Contents2\r\n
    */
}
```



## 실습문제

1. 위에서 문자열을 출력하는 함수 `PutHighlightedString()`은 "H&&ello" 처럼 연속된 &가 사용된 것은 처리하지 않습니다. Esc 절차의 정의에 맞게 동작하도록 `PutHighlightedString()` 함수를 수정하세요.