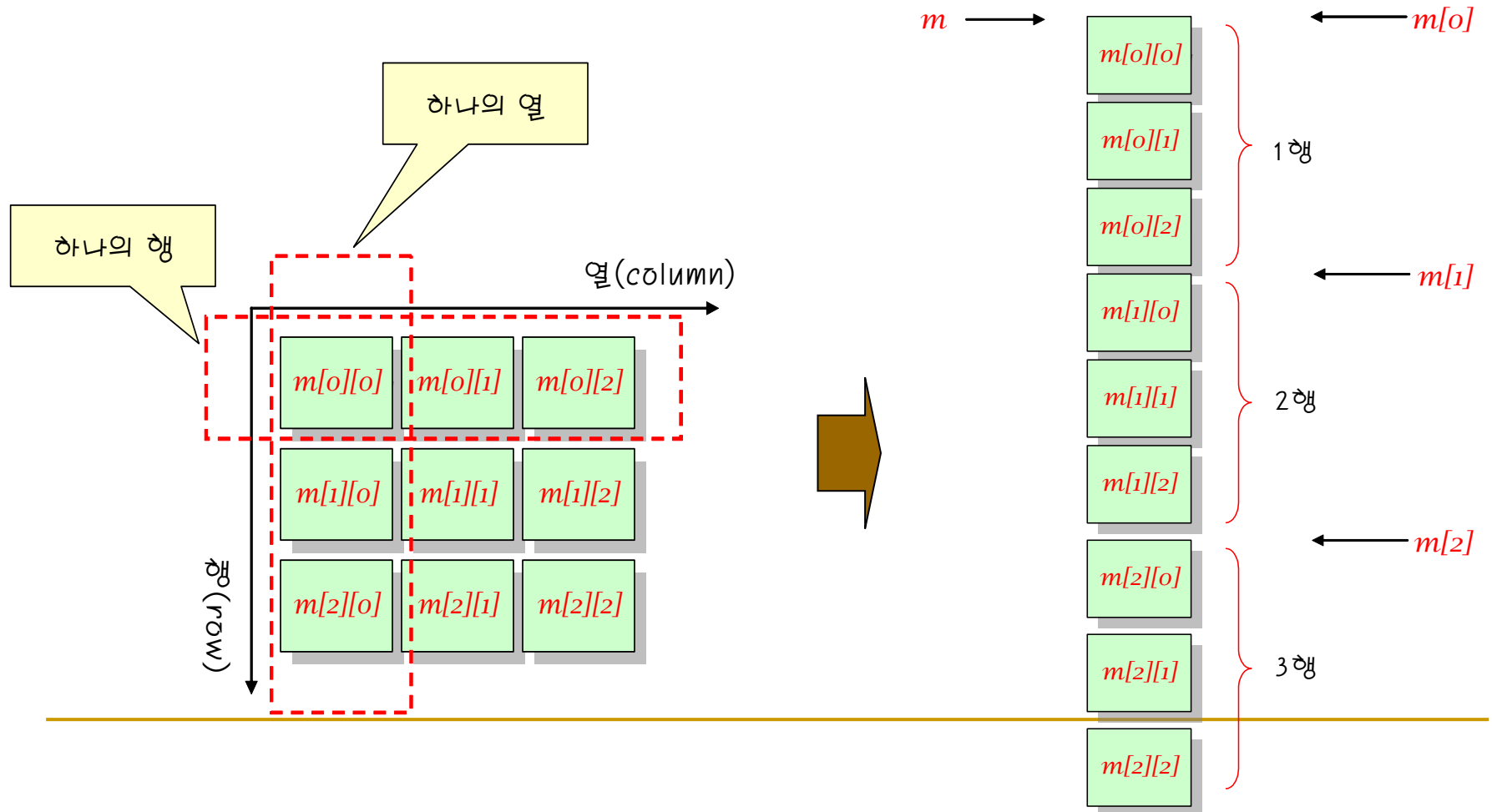


포인터 활용 Part 2

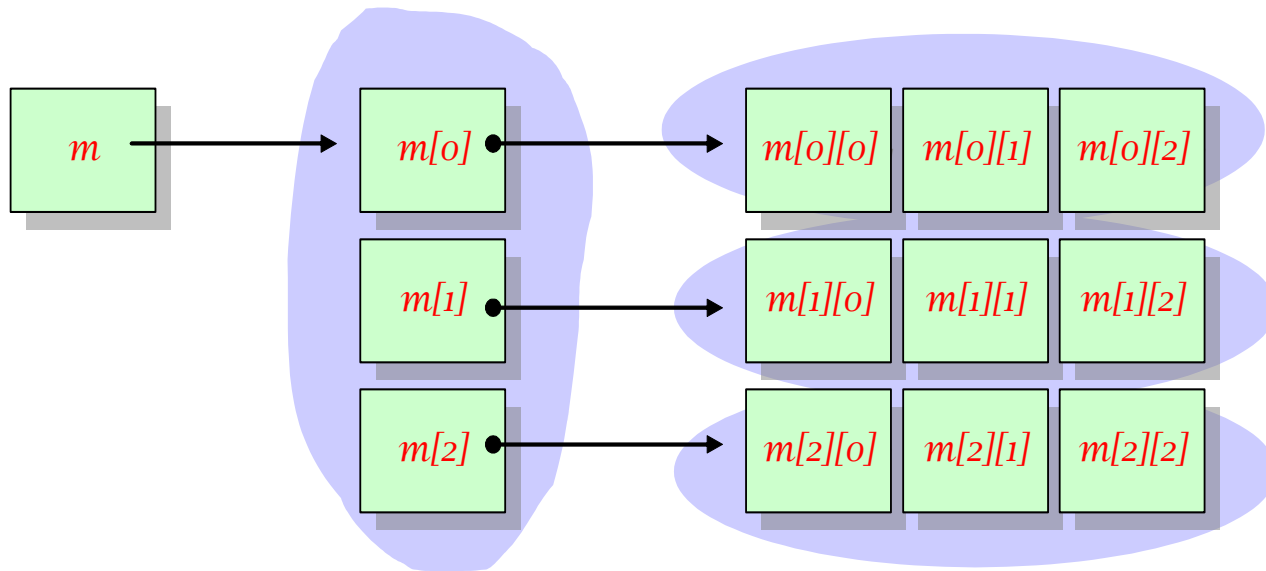
다차원 배열과 포인터

- 2차원 배열 `int m[3][3]`
- 1행→2행→3행→...순으로 메모리에 저장(행우선 방법)



2차원 배열과 포인터

- 배열 이름 m 은 $m[0]=\&m[0][0]$
- $m[0]$ 는 1행의 시작 주소
- $m[1]$ 은 2행의 시작 주소
- ...



multi_array.c

```
// 다차원 배열과 포인터
#include <stdio.h>

int main(void)
{
    int m[3][3] = { 10, 20, 30, 40, 50, 60, 70, 80, 90 };

    printf("m = %p\n", m);
    printf("m[0] = %p\n", m[0]);
    printf("m[1] = %p\n", m[1]);
    printf("m[2] = %p\n", m[2]);
    printf("&m[0][0] = %p\n", &m[0][0]);
    printf("&m[1][0] = %p\n", &m[1][0]);
    printf("&m[2][0] = %p\n", &m[2][0]);

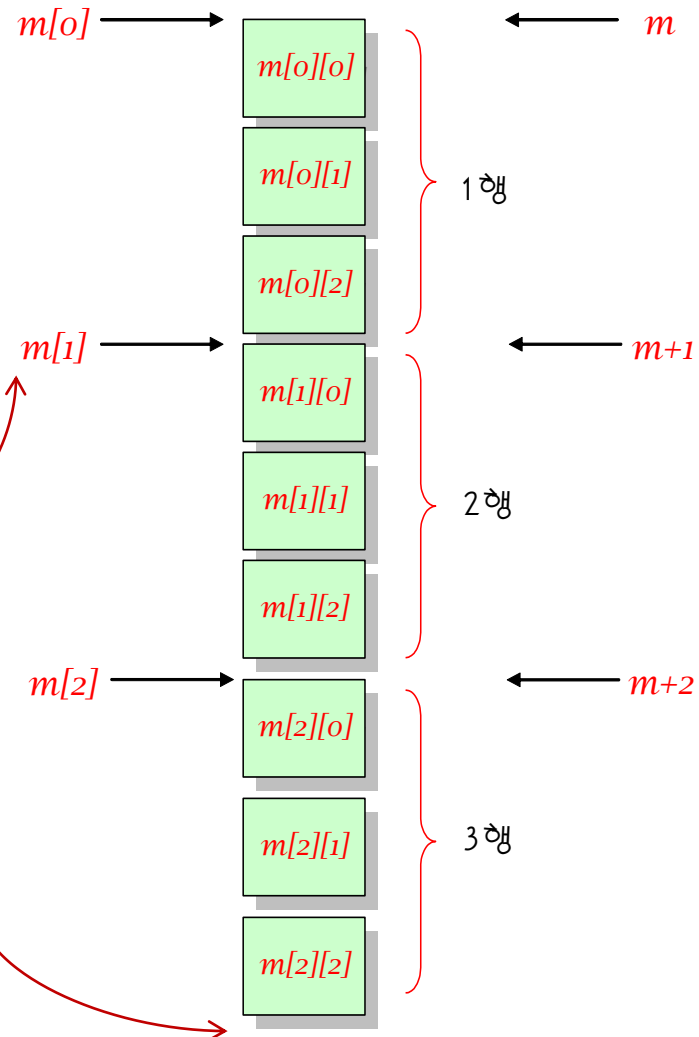
    return 0;
}
```

```
m = 1245020
m[0] = 1245020
m[1] = 1245032
m[2] = 1245044
&m[0][0] = 1245020
&m[1][0] = 1245032
&m[2][0] = 1245044
```

2차원 배열과 포인터 연산

- m 에 대한 연산의 의미
- m 은 $\&m[0][0]$
- $m+1$ 은 $m[1]$
- $m+2$ 은 $m[2]$

- 그러면 $\&m[0]+1=?$
- 그러면 $\&m+1=?$



two_dim_array.c

```
#include <stdio.h>

int main(void)
{
    int m[3][3] = { 10, 20, 30, 40, 50, 60, 70, 80, 90 };

    printf("m = %p\n", m);
    printf("m+1 = %p\n", m+1);
    printf("m+2 = %p\n", m+2);
    printf("m[0] = %p\n", m[0]);
    printf("m[1] = %p\n", m[1]);
    printf("m[2] = %p\n", m[2]);

    return 0;
}
```

```
m = 1245020
m+1 = 1245032
m+2 = 1245044
m[0] = 1245020
m[1] = 1245032
m[2] = 1245044
```

포인터를 이용한 배열 원소 방문

■ 행의 평균을 구하는 경우

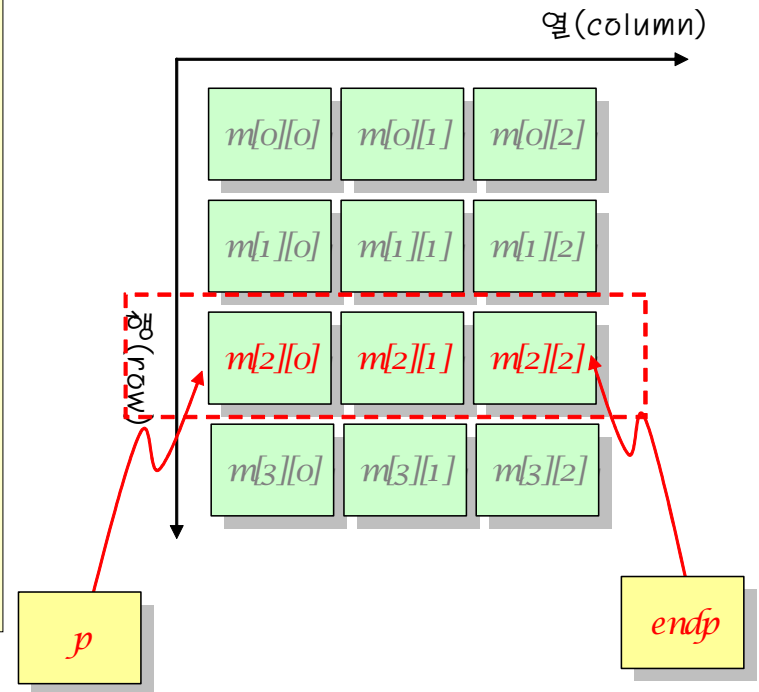
```
double get_row_avg(int m[][COLS], int r)
{
    int *p, *endp;
    double sum = 0.0;

    p = &m[r][0];
    endp = &m[r][COLS];

    while( p < endp )
        sum += *p++;

    sum /= COLS;

    return sum;
}
```



포인터를 이용한 배열 원소 방문

■ 전체 원소의 평균을 구하는 경우

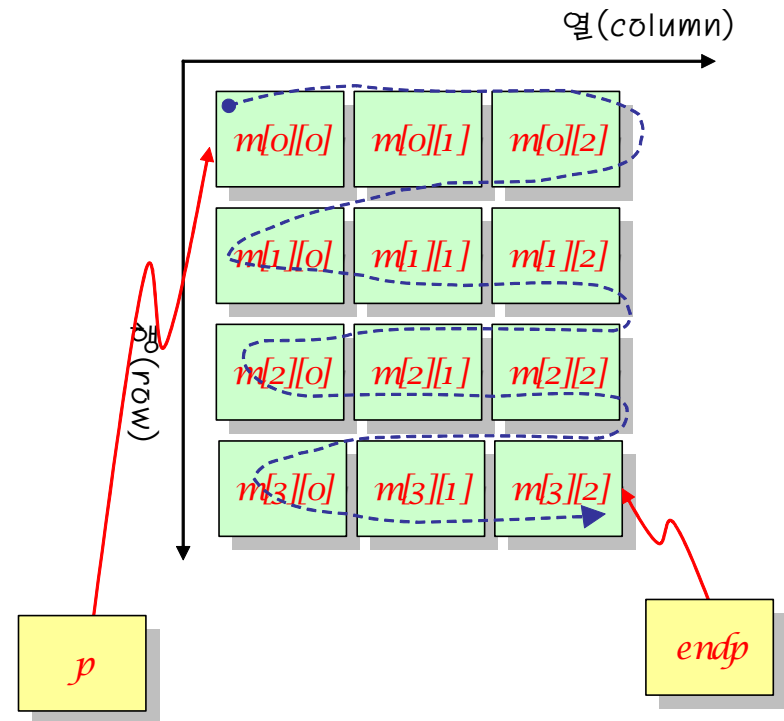
```
double get_total_avg(int m[][COLS])
{
    int *p, *endp;
    double sum = 0.0;

    p = &m[0][0];
    endp = &m[ROWS-1][COLS];

    while( p < endp )
        sum += *p++;

    sum /= ROWS * COLS;

    return sum;
}
```



void 포인터

- 순수하게 메모리의 주소만 가지고 있는 포인터
- 가리키는 대상물은 아직 정해지지 않음

(예) `void *vp;`

- 다음과 같은 연산은 오류이다.

```
*vp;           // 오류  
*(int *)vp;    // void형 포인터를 int형 포인터로 변환한다.  
vp++;          // 오류  
vp--;          // 오류
```



vp.c

```
#include <stdio.h>

int main(void)
{
    int a[] = { 10, 20, 30, 40, 50 };
    void *vp;

    vp = a; // 가능
    vp = &a[2]; // 가능

    *vp = 35; // 오류
    vp++; // 오류

    *(int *)vp = 35; // 가능

    return 0;
}
```

main() 함수의 인수

■ 지금까지의 main() 함수 형태

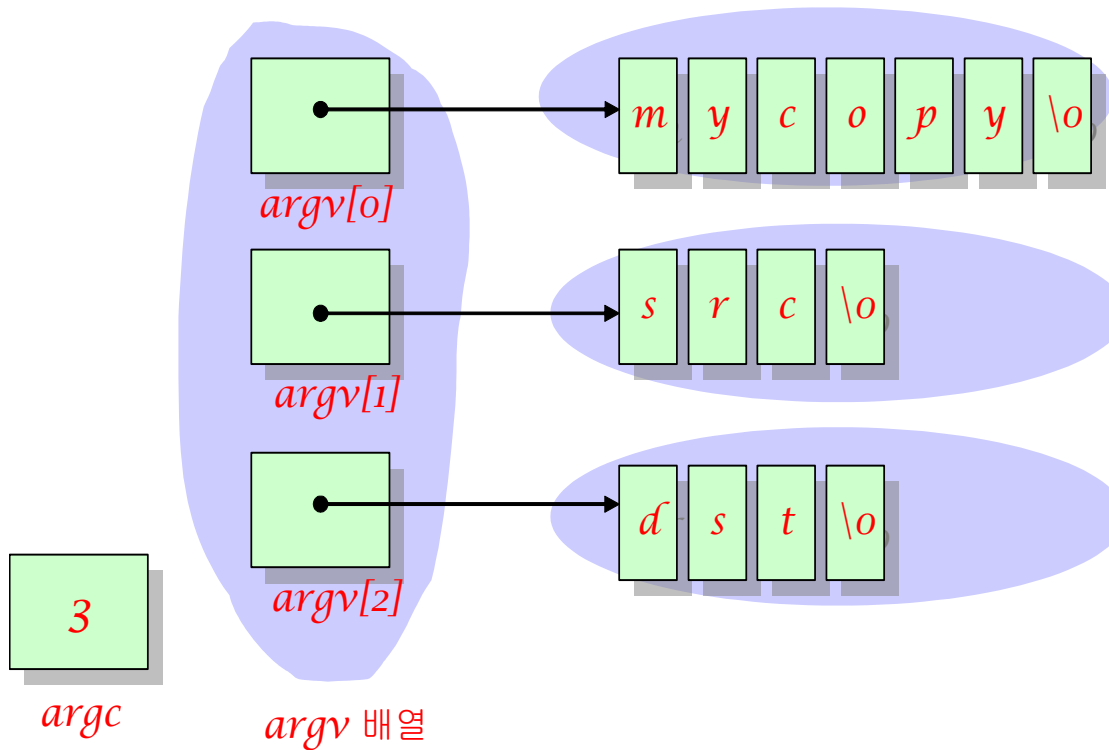
```
int main(void)
{
  ..
}
```

■ 외부로부터 입력을 받는 main() 함수 형태

```
int main(int argc, char *argv[])
{
  ..
}
```

인수 전달 방법

```
C: \cprogram> mycopy src dst
```



main_arg.c

```
#include <stdio.h>

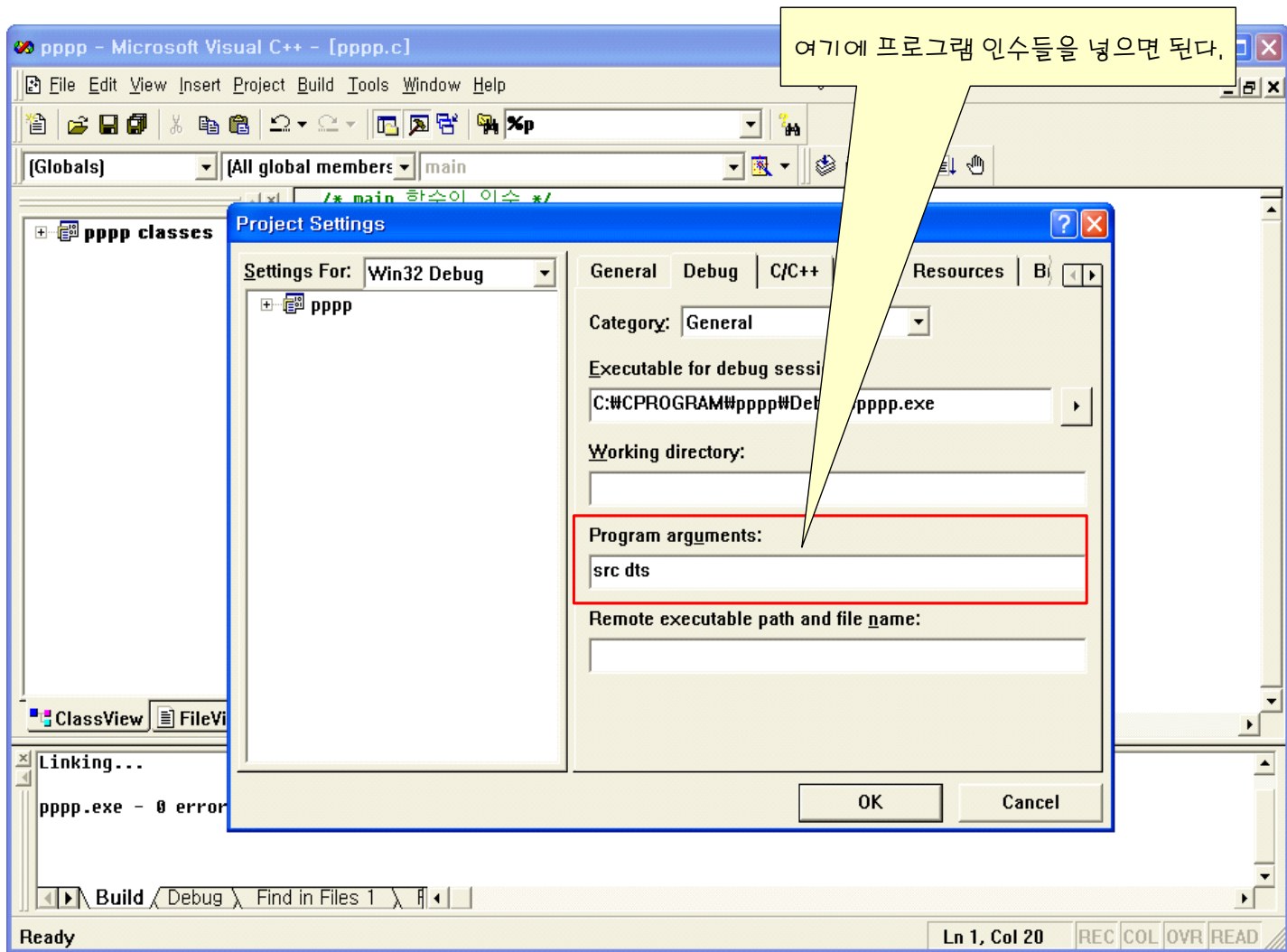
int main(int argc, char *argv[])
{
    int i = 0;

    for(i = 0; i < argc; i++)
        printf("명령어 라인에서 %d번째 문자열 = %s\n", i, argv[i]);

    return 0;
}
```

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
c:\cprogram\mainarg\Debug>mainarg src dst
명령어 라인에서 0번째 문자열 = mainarg
명령어 라인에서 1번째 문자열 = src
명령어 라인에서 2번째 문자열 = dst
c:\cprogram\mainarg\Debug>
```

비주얼 Studio 프로그램 인수 입력



mile2km.c

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

int main(int argc, char *argv[])
{
    double mile, km;

    if( argc != 2 ){
        printf("사용 방법: mile2km 거리\n");
        return 1;
    }
    mile = atof(argv[1]);
    km = 1.609 * mile;
    printf("입력된 거리는 %f km입니다. \n", km);

    return 0;
}
```

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
c:\cprogram\mainarg\Debug>mainarg 10
입력된 거리는 16.090000 km입니다.
c:\cprogram\mainarg\Debug>
```