

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

264

```
int main (int arg C, char * arg V[])
{
    // argument counter.
    // what individual are.
}
```

```
{
return -1; /*error*/
return 0;
}
```

```
gcc -Wall -Wshadow ex1.c -o ex1
// source code -> ex1.c
// output -> ex1
// name of output.
```

★ never have space in filename! ★

```
arg < > = |
arg v [ φ ]
value = ". / ex1" < string
arg v [ 1 ] = " | φ "
```

```
if (arg C < 2)
{
printf ("need a number\n");
return -1; \n new line
}
}
```

```
int main (int arg C,
char * arg v [ φ ])
{
int target; // (unknown)
if ( )
{
return -1;
}
return 0;
}
```

"Student" char []
↑
s, t, u, d, e, n, t

1 char
"string"

7. / ex 1 | φ
argv [1]

" | φ " → int
range = (int) atoi (argv [1], NULL, 10)
string to int

arg v[0] = "x1"

arg v[1] = "10"

"10" → int

~~range~~ argc = 1 → argv[1] does not exist
check argc ≥ 2, so program can run.

```
if ((range < 4) || (range > 16))  
{ printf("range must be between 4 and 15\n");  
  return -1;  
}
```

```
if (a > 5)  
  b = 2  
  c = 2
```

need brackets

it prepares for future
debug.

1/19/11

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

```
{ int row, col;  
  int range;
```

```
  if (argc < 2)
```

```
  {
```

```
    /* error */
```

```
  }
```

```
  range = (int) strtel (argv [1], 0, 10);
```

```
    lex / "3" 5 9
```

```
    argv [0] [1] [2] [3]
```

```
    char
```

```
  if ((range < 4) || (range > 16)) at adds the argument,
```

```
  {
```

```
    /* error */
```

```
    } Logic
```

```
    || &&
```

```
    return -1; OR AND
```

```
  }
```

Not symmetric

short circuit - if a is True, will not check b,
it saves time. If A is false, B will be checked.

```
  if (argc < 2) || (argv [1] [0] == 'c'))  
  for (row = 1; row <= range; row++)
```

```
  {
```

```
    int condition each iteration.
```

```
    printf ("ln") for (col = 1; col <= range; col++)
```

```
    {
```

```
      printf ("%4d", row + col)
```

```
    }
```

```
  }
```

use a function for
- * /

```
void printTable ( int range, int oper )
```

{

```
int row, col;
```

```
for ( row = 1; row <= range; row++
```

{

```
for ( col = 1; col <= range; col++ )
```

{

```
switch ( oper )
```

{

```
case 0; print
```

```
printf ( "%4d", row + col );
```

```
break;
```

```
case 1;
```

```
printf ( "%4d", row - col );
```

```
break;
```

```
default:
```

```
printf ( "Unknown operation");
```

```
/* switch */
```

```
/* col */
```

```
/* row */
```

```
if ( row < 4 || ( row > 16 ) )
```

{

{

```
printTable ( range, 0 )
```

switch = if else
but
cleaner.