



```
*****  
* convolve.c  
***** /
```

```
/* Standard includes */  
#include <assert.h>  
#include <math.h>  
#include <stdlib.h> /* malloc(), realloc() */
```

```
/* Our includes */  
#include "base.h"  
#include "error.h"  
#include "convolve.h"  
#include "klt_util.h" /* printing */
```

```
#define MAX_KERNEL_WIDTH 71
```

```
typedef struct {  
    int width;  
    float data[MAX_KERNEL_WIDTH];  
} ConvolutionKernel;
```

```
/* Kernels */
```

Fundamentals of Programming

session 12

C Program Control, operators precedence

Category	Operator	Associativity
Postfix	() [] -> . ++ --	Left to right
Unary	+ - ! ~ ++ -- (type)* & sizeof	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Shift	<< >>	Left to right
Relational	< <= > >=	Left to right
Equality	== !=	Left to right
Bitwise AND	&	Left to right
Bitwise XOR	^	Left to right
Bitwise OR		Left to right
Logical AND	&&	Left to right
Logical OR		Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %= >>= <<= &= ^= =	Right to left
Comma	,	Left to right

https://www.tutorialspoint.com/cprogramming/c_operators_precedence.htm

operators - comparison

a + b >= c * d

operators - comparison

a > b + c && k == d

operators - comparison

20 > 16 > 10

operators - comparison

$(20 > 16) > 10$

operators - comparison

1 > 10

iterations

Write a program printing squares of 1 to N
(1, 4, 9, ..., N^2)

iterations

```
int N = 10;  
int k;  
  
k = 1;  
while (k <= N) {  
    printf("%d\n",k*k);  
    k++;  
}
```

for loop

```
int N = 10;
int k;

k = 1;
while (k <= N) {
    printf("%d\n",k*k);

    k++;
}
```

```
int N = 10;
int k;

for (k=1; k <= N; k++) {
    printf("%d\n",k*k);
}
```

for loop

```
STATEMENT1;  
while (CONDITION) {  
    // body of loop  
    STATEMENT2;  
}
```

```
for (STATEMENT1; CONDITION; STATEMENT2) {  
    // body of loop  
}
```

for loop

```
for (i = 0, j = N; i <= N; i++,j--) {  
    printf("%d,%d\n",i,j);  
}
```

for loop

```
for (i = 0, j = N; i <= N; i++,j--) {  
    printf("%d,%d\n",i,j);  
}
```

```
for (i = 0, j = N; i <= N; i++,j--) {  
    printf("%2d,%2d\n",i,j);  
}
```

for loop

```
for (i = 0, j = N; i <= N; i++,j--) {  
    printf("%d,%d\n",i,j);  
}
```

```
for (i = 0, j = N; i <= N; i++,j--) {  
    printf("%2d,%2d\n",i,j);  
}
```

Count the ratings

```
int main() {
    int n, r, r1, r2, r3, r4;

    r1 = r2 = r3 = r4 = 0;
    n = 0;
    while (1) {
        printf("Enter rating: ");
        scanf("%d", &r);

        // write your code here
        n++;
    }

    printf("Bad: %.1f%% \n",      100*r1/(float)n);
    printf("Average: %.1f%% \n", 100*r2/(float)n);
    printf("Good: %.1f%% \n",    100*r3/(float)n);
    printf("Excellent: %.1f%% \n", 100*r4/(float)n);

    return 0;
}
```

1: Bad

2: Average

3: Good

4: Excellent

Count the ratings

```
int n, r, r1, r2, r3, r4;

r1 = r2 = r3 = r4 = 0;
n = 0;
while (1) {
    printf("Enter rating: ");
    scanf("%d", &r);
    break;

    if (r== 1)
        r1++;
    else if (r== 2)
        r2++;
    else if (r== 3)
        r3++;
    else if (r== 4)
        r4++;
    else if (r== -1)
        break;
    else {
        puts("invalid number!");
        continue;
    }

    // write your code here
    n++;
}
```

```
finish = 0;
while (1) {
    printf("Enter rating: ");
    scanf("%d", &r);

    switch (r) {
        case 1:
            r1++;
            break;

        case 2:
            r2++;
            break;

        case 3:
            r3++;
            break;

        case 4:
            r4++;
            break;

        case -1:
            finish = 1;
            break;

        default:
            continue;
            break;
    }

    if (finish == 1)
        break;

    n++;
}
```


Count the ratings

```
finish = 0;
while (1) {
    printf("Enter rating: ");
    scanf("%d", &r);

    switch (r) {
        case 1:
            r1++;
            break;

        case 2:
            r2++;
            break;

        case 3:
            r3++;
            break;

        case 4:
            r4++;
            break;

        case -1:
            finish = 1;
            break;

        default:
            continue;
            break;
    }

    if (finish == 1)
        break;

    n++;
}
```

```
finish = 0;
while (1) {
    printf("Enter rating: ");
    scanf("%d", &r);

    switch (r) {
        case 1:
            r1++;
            break;

        case 2:
            r2++;
            break;

        case 3:
            r3++;
            break;

        case 4:
            r4++;
            break;

        case -1:
            finish = 1;
            break;

        default:
            continue;
            break;
    }

    if (finish)
        break;

    n++;
}
```

Count the ratings

```
r12= r34 = 0;
n = 0;

finish = 0;
while (1) {
    printf("Enter rating: ");
    scanf("%d", &r);

    switch (r) {
        case 1:
        case 2:
            r12++;
            break;

        case 3:
        case 4:
            r34++;
            break;

        case -1:
            finish = 1;
            break;

        default:
            continue;
            break;
    }

    if (finish)
        break;

    n++;
}

printf("Bad & Average: %.1f%% \n", 100*r12/(float)n);
printf("Good & Excellent: %.1f%% \n", 100*r34/(float)n);
```

logical operators

`10 < a && a < 16`

logical operators

```
if (|a| > 10)
```

logical operators

`a > 10 || a < -10`

logical operators

```
!(a <= 10 && a >= -10)
```

logical operators

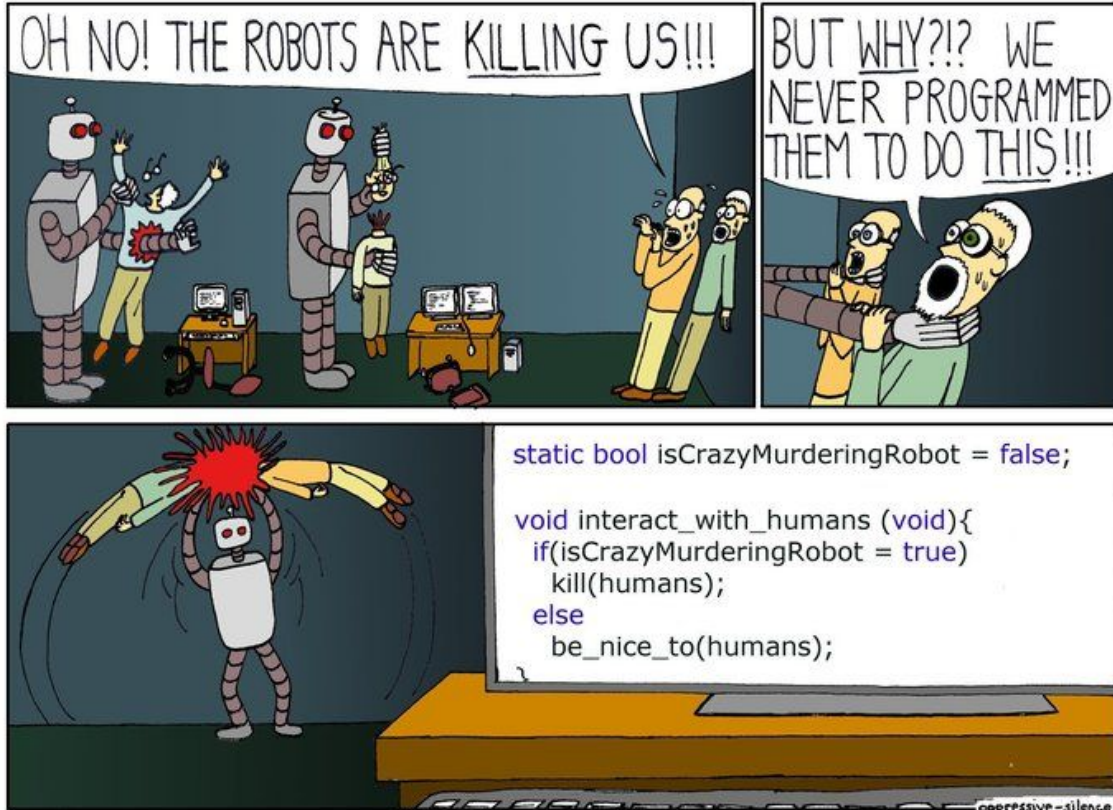
! (a == b) a != b

confusing '==' with '='

```
while {
    safe = check_safe();

    if (safe = 1) {
        demolish_the_building();
        break;
    }
    else
        wait_5_seconds();
}
```


confusing '==' with '='



```
int g,a;

a = 0;
while (1) {
    printf("Enter grade: ");
    scanf("%d", &g);

    if (g== -1)
        break;

    a = a || g < 10;
}

if (a)
    puts("fail");
else
    puts("pass");
```

```
int g,a;

a = 0;
while (1) {
    printf("Enter grade: ");
    scanf("%d", &g);

    if (g== -1)
        break;

    a = a || g < 10;
}

if (a)
    puts("fail");
else
    puts("pass");
```

```
int g,a;

a = 0;
while (1) {
    printf("Enter grade: ");
    scanf("%d", &g);

    if (g== -1)
        break;

    a = a || (g < 10);
}

if (a)
    puts("fail");
else
    puts("pass");
```