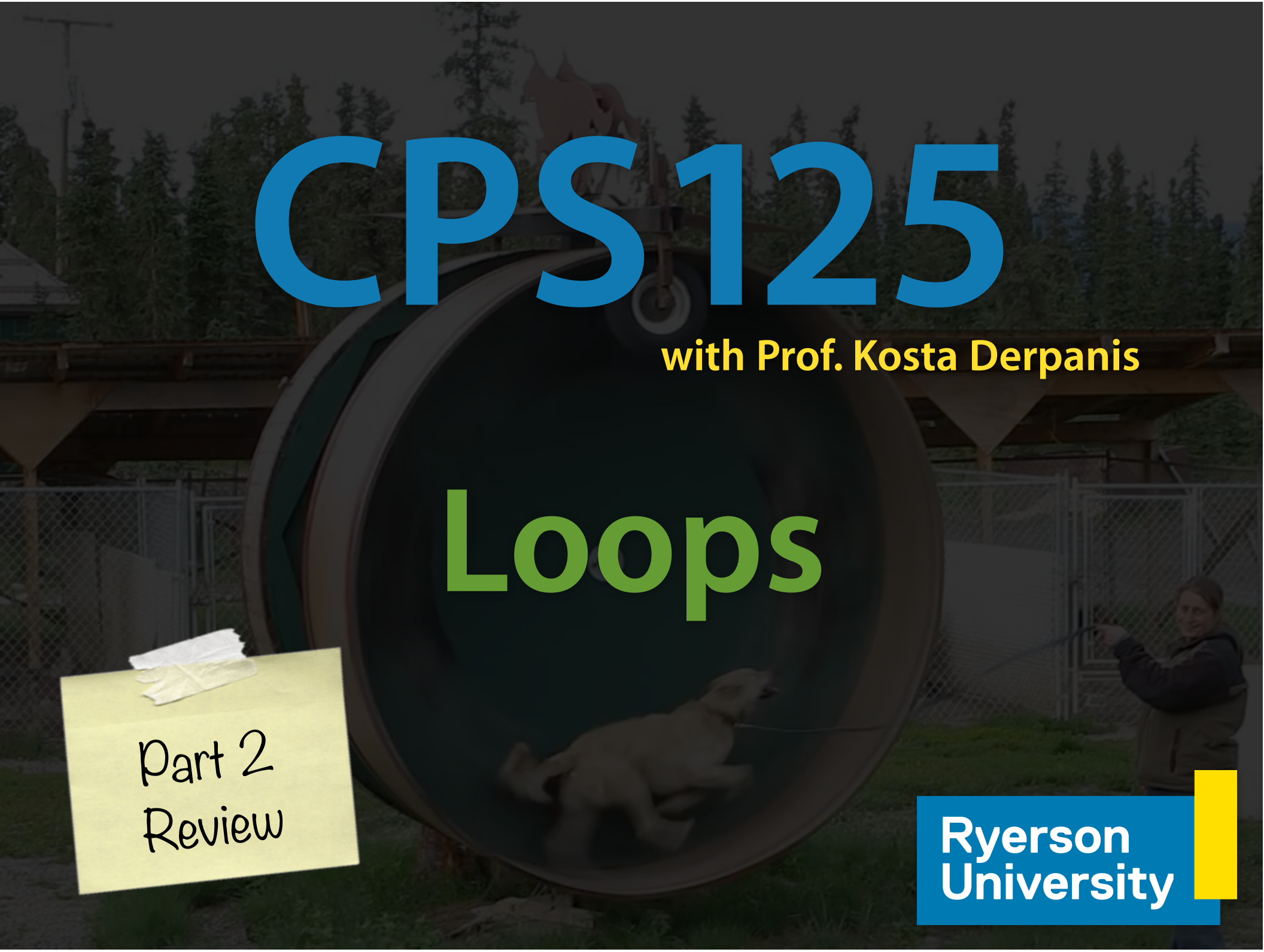
CPS125

with Prof. Kosta Derpanis

Loops

Ryerson University


CPS125
with Prof. Kosta Derpanis
Loops
Part 2
Review
Ryerson
University


Hamelin

Repetition and Loop Statements

CHAPTER 5

CHAPTER OBJECTIVES

- To understand why repetition is an important control structure in programming
- To learn about loop control variables and the three steps needed to control loop repetition
- To learn how to use the C `for`, `while`, and `do-while` statements for writing loops and when to use each statement type
- To learn how to accumulate a sum or a product within a loop body
- To learn common loop patterns such as counting loops, sentinel-controlled loops, and flag-controlled loops
- To understand nested loops and how the outer loop control variable and inner loop control variable are changed in a nested loop
- To learn how to debug programs using a debugger
- To learn how to debug programs by adding diagnostic output statements

Hanly and Koffman

LECTURE TOPICS

LECTURE TOPICS

Loops with file I/O

Loops with File I/O

End of File-controlled loop

loop that reads values from a file and stops when there are no more values left in the file

```
scanf(“%d”, &num1);
```

```
status = scanf("%d", &num1);
```


status = scanf(“%d”, &num1);

number of variables successfully read from file

```
status = scanf("%d", &num1);
```

number of variables successfully read from file

```
status = scanf("%d", &num1);
```

EOF when end of file is reached with input redirection

number of variables successfully read from file

```
status = scanf("%d", &num1);
```

= 0, 1 or **EOF**

EOF when end of file is reached with input redirection

```
/* compute the minimum of the numbers
   in a file using input redirection */

int num, minval = INT_MAX, status;

status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
/* compute the minimum of the numbers
   in a file using input redirection */

int num, minval = INT_MAX, status;

status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
/* compute the minimum of the numbers
   in a file using input redirection */

int num, minval = INT_MAX, status;

status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
/* compute the minimum of the numbers
   in a file using input redirection */

int num, minval = INT_MAX, status;

status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
/* compute the minimum of the numbers
   in a list */

int num, minval = INT_MAX, status;

status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

defined in #include <limits.h>

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
                    39
```

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

22 -1

39

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
39
```

data.txt

```
status = scanf("%d", &num);
```

22

```
while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

22 -1

39

data.txt

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
39
```

data.txt

```
status = scanf("%d", &num);
```

1

```
while (1){
    if (status == EOF) break;

    if (num < minval)
       minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

22 -1

39

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
                    39
```

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
                    39
```

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
       minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
39
```

data.txt

A `break` statement jumps out of the innermost enclosing loop of the statement.

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
       minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
39
```

data.txt

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
39
```

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
39
```

data.txt

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```

39

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```

```
39
```

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```

39

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

-1

```
22 -1
```

39

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```

```
39
```

data.txt

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);

printf("The min num is %d.\n", minval);
```

1

```
22 -1
```

39

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```

39

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
       minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

22 -1

39

data.txt

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

22 -1

39

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

22 -1

39

data.txt

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

22 -1

39

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```

39

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```

39

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

39

```
22 -1
```

39

data.txt

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```

39

data.txt

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);

printf("The min num is %d.\n", minval);
```

1

```
22 -1
```

39

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```


data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
       minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```


data.txt

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```


data.txt

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```


data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```


data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

EOF

```
22 -1
                    39
```

data.txt

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
                    39
```

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
       minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
                    39
```

```
status = scanf("%d", &num);

while (1){
   if (status == EOF) break;

   if (num < minval)
      minval = num;

   status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
                    39
```

```
status = scanf("%d", &num);

while (1){
    if (status == EOF) break;

    if (num < minval)
        minval = num;

    status = scanf("%d", &num);
}

printf("The min num is %d.\n", minval);
```

```
22 -1
```

data.txt


FALSE!
MOLLY
5
BEN
10
SANDRA
25
QUIZ

1. `printf()` and `scanf()` are two functions present in the library `<stdio.h>`.

2. Lines with preprocessor directives in C begin with #.

3. Anything you can do with a `for`-loop can be rewritten using a `while` loop.

4. `while(i =< 10)` statement in C code is correct.

5. Consider the following C code segments a) and b):

a)
```
c = 9;
if(++c >= 10)
    printf("c is greater than 10\n");
```

b)
```
c = 9;
if(c++ >= 10)
    printf("c is greater than 10\n");
```

Both a) and b) would print the statement: `c is greater than 10.`

T

1. `printf()` and `scanf()` are two functions present in the library `<stdio.h>`.

2. Lines with preprocessor directives in C begin with #.

3. Anything you can do with a `for`-loop can be rewritten using a `while` loop.

4. `while(i =< 10)` statement in C code is correct.

5. Consider the following C code segments a) and b):

a)
```
c = 9;
if(++c >= 10)
   printf("c is greater than 10\n");
```

b)
```
c = 9;
if(c++ >= 10)
   printf("c is greater than 10\n");
```

Both a) and b) would print the statement: `c is greater than 10.`

T 1. `printf()` and `scanf()` are two functions present in the library `<stdio.h>`.

T 2. Lines with preprocessor directives in C begin with #.

3. Anything you can do with a `for`-loop can be rewritten using a `while` loop.

4. `while(i =< 10)` statement in C code is correct.

5. Consider the following C code segments a) and b):

a)
```
c = 9;
if(++c >= 10)
    printf("c is greater than 10\n");
```

b)
```
c = 9;
if(c++ >= 10)
    printf("c is greater than 10\n");
```

Both a) and b) would print the statement: `c is greater than 10.`

T 1. `printf()` and `scanf()` are two functions present in the library `<stdio.h>`.

T 2. Lines with preprocessor directives in C begin with #.

T 3. Anything you can do with a `for`-loop can be rewritten using a `while` loop.

4. `while(i =< 10)` statement in C code is correct.

5. Consider the following C code segments a) and b):

a)
```
c = 9;
if(++c >= 10)
    printf("c is greater than 10\n");
```

b)
```
c = 9;
if(c++ >= 10)
    printf("c is greater than 10\n");
```

Both a) and b) would print the statement: `c is greater than 10.`

T 1. `printf()` and `scanf()` are two functions present in the library `<stdio.h>`.

T 2. Lines with preprocessor directives in C begin with #.

T 3. Anything you can do with a `for`-loop can be rewritten using a `while` loop.

F 4. `while(i =< 10)` statement in C code is correct.

5. Consider the following C code segments a) and b):

a)
```
c = 9;
if(++c >= 10)
    printf("c is greater than 10\n");
```

b)
```
c = 9;
if(c++ >= 10)
    printf("c is greater than 10\n");
```

Both a) and b) would print the statement: `c is greater than 10.`

T 1. `printf()` and `scanf()` are two functions present in the library `<stdio.h>`.

T 2. Lines with preprocessor directives in C begin with #.

T 3. Anything you can do with a `for`-loop can be rewritten using a `while` loop.

F 4. `while(i =< 10)` statement in C code is correct.

5. Consider the following C code segments a) and b):

a)
```
c = 9;
if(++c >= 10)
    printf("c is greater than 10\n");
```

F

b)
```
c = 9;
if(c++ >= 10)
    printf("c is greater than 10\n");
```

Both a) and b) would print the statement: `c is greater than 10.`

Identify the incorrect statements:

```
(a) if(a = 5)
(b) DOUBLE = 10
(c) a = 5+-a
(d) Int = 5 + 2
(e) a ** 5
(f) x = 3y
(g) int *a
(h) int* a
(i) printf/7
(j) _a = x * -2
(k) car8 = a - 5 /2
(l) one + two + three
(m) A5A * !!!!4
(n) char c = 10
```

Identify the incorrect statements:

(a) `if(a = 5)`

(b) `DOUBLE = 10`

(c) `a = 5+-a`

(d) `Int = 5 + 2`

(e) `a ** 5`

(f) `x = 3y`

(g) `int *a`

(h) `int* a`

(i) `printf/7`

(j) `_a = x * -2`

(k) `car8 = a - 5 /2`

(l) `one + two + three`

(m) `A5A * !!!!4`

(n) `char c = 10`

Convert the following binary number (unsigned 32-bit integers) to decimal:

```
0000 0000 0000 0000 0000 0000 0011 0110
(a) 63
(b) 55
(c) 54
(d) 8
(e) -13
```

Convert the following binary number (unsigned 32-bit integers) to decimal:

```
0000 0000 0000 0000 0000 0000 0011 0110
(a) 63
(b) 55
(c) 54
(d) 8
(e) -13
```

Convert the following binary number (unsigned 32-bit integers) to decimal:

```
0000 0000 0000 0000 0000 0000 0011 0110
(a) 63
(b) 55
(c) 54
(d) 8
(e) -13
```

even number

Convert the following binary number (unsigned 32-bit integers) to decimal:

```
0000 0000 0000 0000 0000 0000 0011 0110
```

even number

(a) 63

(b) 55

(c) 54

(d) 8

(e) -13

Convert the following binary number (unsigned 32-bit integers) to decimal:

0000 0000 0000 0000 0000 0000 0011 0110

even number

(a) 63

(b) 55

(c) 54

(d) 8

(e) 13

Convert the following decimal number to 32-bit two's complement:

63

(a) 1000 0000 0000

(b) 1111 0000 0000 0000 0000 1110 0000 0001

(c) 0000 0000 0000 0000 0000 0000 0011 1111

(d) 0000 0000 0000 0000 0000 0000 0100 0000

Convert the following decimal number to 32-bit two's complement:

63

(a) 1000 0000 0000 **not 32-bit**

(b) 1111 0000 0000 0000 0000 1110 0000 0001

(c) 0000 0000 0000 0000 0000 0000 0011 1111

(d) 0000 0000 0000 0000 0000 0000 0100 0000

Convert the following decimal number to 32-bit two's complement:

63

(a) 1000 0000 0000 0000

(b) 1111 0000 0000 0000 0000 1110 0000 0001

(c) 0000 0000 0000 0000 0000 0000 0011 1111

(d) 0000 0000 0000 0000 0000 0000 0100 0000

Convert the following decimal number to 32-bit two's complement:

63

(a) 1000 0000 0000 0000

(b) 1111 0000 0000 0000 0000 0000 0000 0000

negative number

(c) 0000 0000 0000 0000 0000 0000 0011 1111

(d) 0000 0000 0000 0000 0000 0000 0100 0000

Convert the following decimal number to 32-bit two's complement:

63

(a) 1000 0000 0000 0000

(b) 1111 0000 0000 0000 0000 0000 1110 0000 0001

(c) 0000 0000 0000 0000 0000 0000 0011 1111

(d) 0000 0000 0000 0000 0000 0000 0100 0000

Convert the following decimal number to 32-bit two's complement:

63

(a) 1000 0000 0000

(b) 1111 0000 0000 0000 0000 0000 1110 0000 0001

(c) 0000 0000 0000 0000 0000 0000 0011 1111

(d) 0000 0000 0000 [illegible] 0 0100 0000

even number

Convert the following decimal number to 32-bit two's complement:

63

(a) 1000 0000 0000 0000

(b) 1111 0000 0000 0000 0000 0000 1110 0000 0001

(c) 0000 0000 0000 0000 0000 0000 0011 1111

(d) 0000 0000 0000 0000 0000 0000 0100 0000

Convert the following decimal number to 32-bit two's complement:

-7

(a) 0000 0000 0000 0000 0000 0000 0000 0111

(b) 1111 0000 0000 0000 0000 1110 0000 0001

(c) 0000 0000 0000 0000 0000 0000 0011 1111

(d) 1111 1111 1111 1111 1111 1111 1111 1001

Convert the following decimal number to 32-bit two's complement:

-7

(a) 0000 0000 0000 0000 0000 0000 0000 0111

(b) 1111 0000 0000 0000 0000 1110 0000 0001

(c) 0000 0000 0000 0000 0000 0000 0011 1111

(d) 1111 1111 1111 1111 1111 1111 1111 1001

Convert the following decimal number to 32-bit two's complement:

-7

(a) 0000 0000 0000 0000 0000 0000 0000 0111

(b) 1111 0000 0000 0000 0000 1110 0000 0001

(c) 0000 0000 0000 0000 0000 0000 0011 1111

(d) 1111 1111 1111 1111 1111 1111 1111 1001

Convert the following decimal number to 32-bit two's complement:

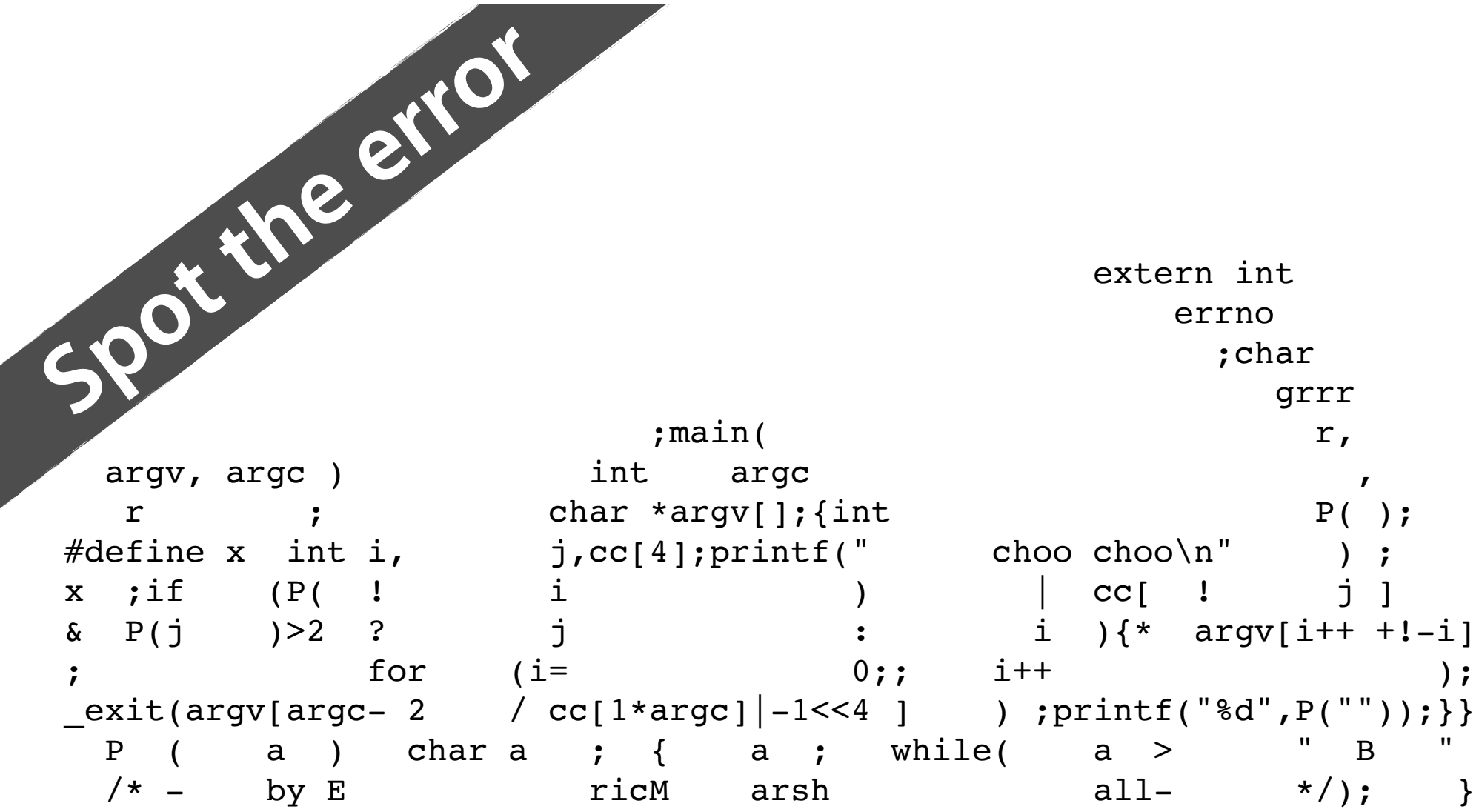
-7

(a) 0000 0000 0000 0000 0000 0000 0000 0111

(b) 1111 0000 0000 0000 0000 1110 0000 0001

(c) 0000 0000 0000 0000 0000 0000 0011 1111

(d) 1111 1111 1111 1111 1111 1111 1111 1001


Spot the error
extern int
errno
;char
grrr
r,
,
P( );
) ;
j]
argv[i++ +!-i]
);
;printf("%d",P(""));}}
" B "
*/); }
;main(
int argc
char *argv[];{int
j,cc[4];printf("
)
:
0;;
i
){* argv, argc)
r ;
#define x int i,
x ;if (P(!
& P(j)>2 ?
;
for (i=
/ cc[1*argc]|-1<<4]
_exit(argv[argc- 2
P (a) char a ; { a ; while(
/* - by E
ricM arsh
choo choo\n"
| cc[!
i
++
a >
all-

```
                                                    extern int
                                                        errno
                                                          ;char
                                                             grrr
                                                               r,
                                  ;main(                         ,
  argv, argc )                      int    argc                  P( );
   r         ;                    char *argv[];{int      choo choo\n"    ) ;
#define x  int i,                  j,cc[4];printf("        |  cc[  !     j ]
x  ;if    (P(  !                   i              )       i  ){*  argv[i++ +!-i]
&  P(j    )>2  ?                   j              :     i++                    );
;             for    (i=           0;;    cc[1*argc]|-1<<4 ]    ) ;printf("%d",P(""));}}
_exit(argv[argc- 2    / cc[1*argc]|-1<<4 ]    ; while(    a  >      "  B   "
  P  (    a  )   char a   ; {    a  ;          arsh            all-     */);    }
  /* -    by E           ricM
```

The International Obfuscated C Code Contest

[The judges | IOCCC home page | How to enter | FAQ |
IOCCC news | People who have won | **Winning entries**]

[2013 | 2012 | 2011 | 2006 | 2005 | 2004 | 2003 | 2002 | 2001 | 2000 | 1999 | 1998 | 1997 | 1996 |
1995 | 1994 | 1993 | 1992 | 1991 | 1990 | 1989 | 1988 | 1987 | 1986 | 1985 | 1984 | ALL]

Winning Entries

This version of the winning entries page does **NOT** contain brief explanations of each winning entry. You can check out the **spoiler** version of this page if you don't mind seeing a brief summary of each entry. There is also a spoiler summary that gives a summary description of all of the winners.

22nd International Obfuscated C Code Contest (2013)

- 2013.tar.bz2 - all of the data for 2013
- Makefile
- README.html
- README.text
- hint.css
- whowon.html

birken

ioccc.org/years.html

PRIVATE Reader

Previous IOCCC Winners

The International Obfuscated C Code Contest

[The judges | IOCCC home page | How to enter | FAQ |
IOCCC news | People who have won | **Winning entries**]

[2013 | 2012 | 2011 | 2006 | 2005 | 2004 | 2003 | 2002 | 2001 | 2000 | 1999 | 1998 | 1997 | 1996 |
1995 | 1994 | 1993 | 1992 | 1991 | 1990 | 1989 | 1988 | 1987 | 1986 | 1985 | 1984 | ALL]

Winning Entries

This version of the winning entries page does **NOT** contain brief explanations of each winning entry. You can check out the **spoiler** version of this page if you don't mind seeing a brief summary of each entry. There is also a spoiler summary that gives a summary description of all of the winners.

22nd International Obfuscated C Code Contest (2013)

- 2013.tar.bz2 - all of the data for 2013
- Makefile
- README.html
- README.text
- hint.css
- whowon.html

birken

Spot the error

```
#include <stdio.h>

int main (void)
{
    int x = 5;
    while( x > 0 );
        x--;

    return 55;
}
```

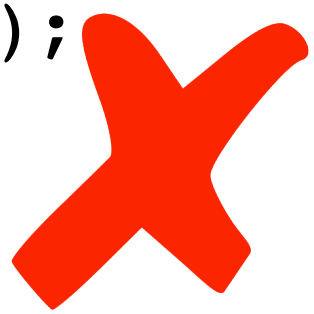
```
#include <stdio.h>

int main (void)
{
    int x = 5;
    while( x > 0 );
        x--;

    return 55;
}
```

```
#include <stdio.h>

int main (void)
{
    int x = 5;
    while( x > 0 );
        x--;

    return 55;
}
```

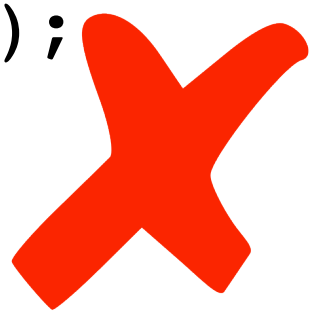
Spot the error

```
#include <stdio.h>

int main (void)
{
      float number;

      if (-100.1)
         printf("Enter a number: ");

      scanf("%f", &number);

      if (number > 100)
         printf("The input is ");
         printf("> 100\n");
      else
         printf("The input is <= 100");

      return(0);
}
```

```
#include <stdio.h>

int main (void)
{
     float number;

     if (-100.1)
        printf("Enter a number: ");

     scanf("%f", &number);

     if (number > 100)
        printf("The input is ");
        printf("> 100\n");
     else
        printf("The input is <= 100");

     return(0);
}
```



```
#include <stdio.h>

int main (void)
{
      float number;

      if (-100.1)
         printf("Enter a number: ");

      scanf("%f", &number);

      if (number > 100) {
         printf("The input is ");
         printf("> 100\n");}
      else
         printf("The input is <= 100");

      return(0);
}
```



Spot the error

```
#include <stdio.h>

int main (void)
{
   float a = 10; int b = 10;

   switch(a)
   {
      case b:

         printf("a == 10\n");


      case 20:
         printf("a == 20\n");

   }
   return 1;
}
```

```
#include <stdio.h>

int main (void)
{
   float a = 10; int b = 10;

   switch(a)
   {
      case b:

         printf("a == 10\n");


      case 20:
         printf("a == 20\n");

   }
   return 1;
}
```

```
#include <stdio.h>

int main (void)
{
    float a = 10; int b = 10;

    switch(a)
    {
        case b:

            printf("a == 10\n");


        case 20:
            printf("a == 20\n");

    }
    return 1;
}
```

```
#include <stdio.h>

int main (void)
{
    switch(a)
    {
        case b:

            printf("a == 10\n");


        case 20:
            printf("a == 20\n");

    }
    return 1;
}
```

control value type can only be `int` or `char`

```
#include <stdio.h>

int main (void)
{
    float a = 10; int b = 10;

    switch(a)
    {
        case b:
            pri

        case 20:
            printf("a == 20\n");

    }
    return 1;
}
```

case expressions must be constant

```
#include <stdio.h>

int main (void)
{
   float a = 10; int b = 10;

   switch(a)
   {
      case b:

         printf("a == 10\n");
         break;

      case 20:
         printf("a == 20\n");

   }
   return 1;
}
```

```
#include <stdio.h>

int main (void)
{
    float a = 10; int b = 10;

    switch(a)
    {
        case b:

            printf("a == 10\n");
            break;

        case 20:
            printf("a == 20\n");
            break;
    }
    return 1;
}
```

optional

Spot the error

```
#include <stdio.h>

int main (void)
{
   int temp;

   scanf("Enter a temperature %d",
         &temp);

   if (temp < 0)
      printf("It's cold\n");
   else if (0 <= temp <= 25)
      printf("It's warm\n");
   else
        printf("It's HOT!!!\n");

     return(0);
}
```

```
#include <stdio.h>

int main (void)
{
   int temp;

   scanf("Enter a temperature %d",
         &temp);

   if (temp < 0)
      printf("It's cold\n");
   else if (0 <= temp <= 25)
      printf("It's warm\n");
   else
         printf("It's HOT!!!\n");

     return(0);
}
```

```
#include <stdio.h>

int main (void)
{
    int temp;

    scanf("Enter a temperature %d",
          &temp);

    if (temp < 0)
       printf("It's cold\n");
    else if (0 <= temp <= 25)
       printf("It's warm\n");
    else
         printf("It's HOT!!!\n");

      return(0);
}
```

```
#include <stdio.h>

int main (void)
{
    int temp;

    scanf("Enter a temperature %d",
          &temp);

    if (temp < 0)
       printf("It's cold\n");
    else if (0 <= temp <= 25)
       printf("It's ...
    else
          printf("It's HOT!!!\n");

      return(0);
}
```

Logical error always true

```
#include <stdio.h>

int main (void)
{
    int temp;

    scanf("Enter a temperature %d",
          &temp);

    if (temp < 0)
       printf("It's cold\n");
    else if (0 <= temp <= 25)
       printf("
    else
          printf("It's HOT!!!\n");

      return(0);
}
```

Logical error always true

Spot the error

```
#include <stdio.h>

int main(void)
{
    int i;

    i = 1;

    for (; i <= 12; i = i + 1 )
    {
        printf("%d\n",i);
    }

    return (0);
}
```

```c
#include <stdio.h>

int main(void)
{
    int i;

    i = 1;

    for (; i <= 12; i = i + 1 )
    {
        printf("%d\n",i);
    }

    return (0);
}
```

```c
#include <stdio.h>

int main(void)
{
    int i;

    i = 1;

    for (; i <= 12; i = i + 1 )
    {
        printf("%d\n",i);
    }

    return (0);
}
```

Spot the error

```
#include <stdio.h>

int main(void)
{

    for (i = 1; i <= 12; i++)
    {
       printf("%d\n",i);
    }

    return (0);
}
```

```
#include <stdio.h>

int main(void)
{

    for (i = 1; i <= 12; i++)
    {
        printf("%d\n",i);
    }

    return (0);
}
```

```
#include <stdio.h>

int main(void)
{
    for (i = 1; i <= 12; i++)
    {
        printf("%d\n",i);
    }

    return (0);
}
```

variable not declared

Spot the error

```
#include <stdio.h>

int main(void)
{
    int n;

    do
    {
        printf("Enter a number ");
        printf("between 1 and 9: ");
        scanf ("%d", &n);
    }while (n < 1 || n > 9)

 return (0);
}
```

```c
#include <stdio.h>

int main(void)
{
    int n;

    do
    {
        printf("Enter a number ");
        printf("between 1 and 9: ");
        scanf ("%d", &n);
    }while (n < 1 || n > 9)

  return (0);
}
```

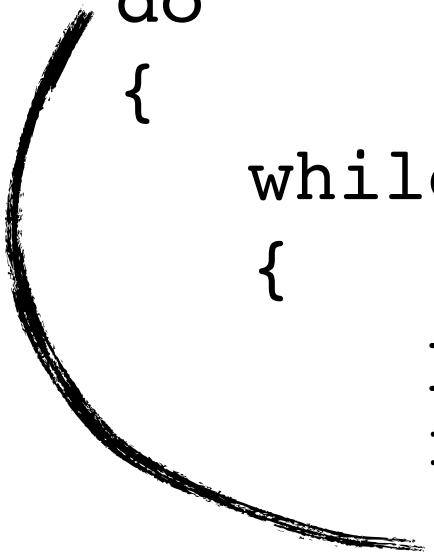
```c
#include <stdio.h>

int main(void)
{
    int n;

    do
    {
        printf("Enter a number ");
        printf("between 1 and 9: ");
        scanf ("%d", &n);
    }while (n < 1 || n > 9);

  return (0);
}
```

Spot the error

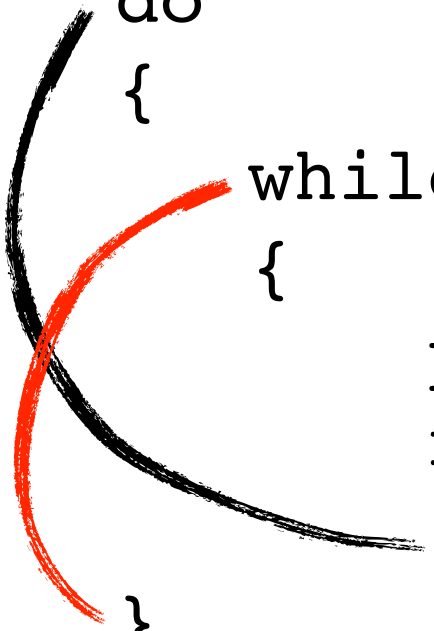
```
#include <stdio.h>

int main(void)
{
    int n = 1;

    do
    {
       while (n <= 2)
       {
           printf("%d\n", n);
           n = n + 1;
           } while (n <= 2);
    }
    return (0);
}
```

```
#include <stdio.h>

int main(void)
{
    int n = 1;

    do
    {
       while (n <= 2)
       {
           printf("%d\n", n);
           n = n + 1;
           } while (n <= 2);
    }
    return (0);
}
```

```c
#include <stdio.h>

int main(void)
{
    int n = 1;

    do
    {
        while (n <= 2)
        {
            printf("%d\n", n);
            n = n + 1;
        } while (n <= 2);
    }
    return (0);
}
```



```c
#include <stdio.h>

int main(void)
{
    int n = 1;

    do
    {
        while (n <= 2)
        {
            printf("%d\n", n);
            n = n + 1;
        } while (n <= 2);
    }
    return (0);
}
```



```
double a = 3, b = 4;

printf ("%6.3lf", (int)a/b);
```

```
double a = 3, b = 4;

printf ("%6.3lf", (int)a/b);
```

What is the output?

```
double a = 3, b = 4;

printf ("%6.3lf", (int)a/b);
```

Console

```
   #0.750
```

```
double a = 3, b = 4;

printf ("%6.3lf", (int)a/b);
```

Console

```
   #0.750
```

Space is indicated by #

```
double a = 3, b = 4, c = 1, e = -2;

printf ("-%d%d%d%d", a>2,b<5,c==2,!e);
```

```
double a = 3, b = 4, c = 1, e = -2;

printf ("-%d%d%d%d", a>2,b<5,c==2,!e);
```

What is the output?

```
double a = 3, b = 4, c = 1, e = -2;

printf ("-%d%d%d%d", a>2,b<5,c==2,!e);
```

Console

```
-1100
```

```
double a = 9.5;
int b = 3;
int *c = &b;

printf ("%d", (int)a + -*c);
```

```
double a = 9.5;
int b = 3;
int *c = &b;

printf ("%d", (int)a + -*c);
```

What is the output?

```
double a = 9.5;
int b = 3;
int *c = &b;

printf ("%d", (int)a + -*c);
```

Console

```
6
```

```
for (i = 0; i < 10; ++i)
   printf ("%d ", i++);

printf ("%d", i);
```

```
for (i = 0; i < 10; ++i)
   printf ("%d ", i++);

printf ("%d", i);
```

What is the output?

```
for (i = 0; i < 10; ++i)
   printf ("%d ", i++);

printf ("%d", i);
```

Console

```
0 2 4 6 8 10
```

```
for (i = 50; i >= 0; i = i - 1);
   printf("i = %d\n", i);

while (i <= 50);
{
   printf("i = %d\n", i++);
}

printf("Program complete!\n");
```

```
for (i = 50; i >= 0; i = i - 1);
   printf("i = %d\n", i);

while (i <= 50);
{
   printf("i = %d\n", i++);
}

printf("Program complete!\n");
```

What's the output?

```
for (i = 50; i >= 0; i = i - 1);
   printf("i = %d\n", i);

while (i <= 50);
{
   printf("i = %d\n", i++);
}

printf("Program complete!\n");
```

```
for (i = 50; i >= 0; i = i - 1);
   printf("i = %d\n", i);

while (i <= 50);
{
   printf("i = %d\n", i++);
}

printf("Program complete!\n");
```

i = -1

```
for (i = 50; i >= 0; i = i - 1);
   printf("i = %d\n", i);
```

```
while (i <= 50);
{
   printf("i = %d\n", i++);
}

printf("Program complete!\n");
```

```
for (i = 50; i >= 0; i = i - 1);
   printf("i = %d\n", i);
```

```
while (i <= 50);
{
   printf("i = %d\n", i++);
}

printf("Program complete!\n");
```

no output, infinite loop

```
for (i = 50; i >= 0; i = i - 1);
   printf("i = %d\n", i);

while (i <= 50);
{
}

printf("Program complete!\n");
```

no output, infinite loop

```
for (i = 50; i >= 0; i = i - 1);
   printf("i = %d\n", i);

while (i <= 50);
{
   printf("i = %d\n", i++);
}

printf("Program complete!\n");
```

Write a code snippet that reads in an integer from the keyboard and adds up its digits.

Solution

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
   remainder = t % 10;
   sum       = sum + remainder;
   t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
   n, sum);
```

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
   remainder = t % 10;
   sum       = sum + remainder;
   t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
   n, sum);
```

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
   remainder = t % 10;
   sum       = sum + remainder;
   t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
   n, sum);
```

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
   remainder = t % 10;
   sum       = sum + remainder;
   t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
   n, sum);
```

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
   remainder = t % 10;
   sum       = sum + remainder;
   t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
   n, sum);
```

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
    remainder = t % 10;
    sum       = sum + remainder;
    t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
    n, sum);
```

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
    remainder = t % 10;
    sum       = sum + r
    t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
    n, sum);
```

get rightmost digit

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
   remainder = t % 10;
   sum       = sum + remainder;
   t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
   n, sum);
```

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
    remainder = t % 10;
    sum       = sum + remainder;
    t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
    n, sum);
```

add to running sum of digits

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
   remainder = t % 10;
   sum       = sum + remainder;
   t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
   n, sum);
```

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
   remainder = t % 10;
   sum       = sum + remainder;
   t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
   n, sum);
```

remove rightmost digit

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
   remainder = t % 10;
   sum       = sum + remainder;
   t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
   n, sum);
```

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
   remainder = t % 10;
   sum       = sum + remainder;
   t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
   n, sum);
```

Solution

```
printf("Enter an integer\n");
scanf("%d", &n);

t   = n;
sum = 0;

while (t != 0)
{
   remainder = t % 10;
   sum       = sum + remainder;
   t         = t / 10;
}

printf("Sum of digits of %d = %d\n",
   n, sum);
```

Write a code snippet that reads integers from a file called input.txt and only adds up all the negative numbers.

The number of integers in the file is unknown.

Solution

```
int number, sum = 0, status;
FILE* in;

in  = fopen("input.txt", "r");
status = fscanf(in, "%d", &number);

while (status != EOF)
{
   if (number < 0)
   {
      sum = sum + number;
   }
   status = fscanf(in, "%d", &number);
}

printf("The sum is %d.\n", sum);
fclose(in);
```

```
int number, sum = 0, status;
FILE* in;

in  = fopen("input.txt", "r");
status = fscanf(in, "%d", &number);

while (status != EOF)
{
   if (number < 0)
   {
      sum = sum + number;
   }
   status = fscanf(in, "%d", &number);
}

printf("The sum is %d.\n", sum);
fclose(in);
```

```
int number, sum = 0, status;
FILE* in;

in  = fopen("input.txt", "r");
status = fscanf(in, "%d", &number);

while (status != EOF)
{
   if (number < 0)
   {
      sum = sum + number;
   }
   status = fscanf(in, "%d", &number);
}

printf("The sum is %d.\n", sum);
fclose(in);
```

```
int number, sum = 0, status;
FILE* in;

in  = fopen("input.txt", "r");
status = fscanf(in, "%d", &number);

while (status != EOF)
{
   if (number < 0)
   {
      sum = sum + number;
   }
   status = fscanf(in, "%d", &number);
}

printf("The sum is %d.\n", sum);
fclose(in);
```

```
int number, sum = 0, status;
FILE* in;

in  = fopen("input.txt", "r");
status = fscanf(in, "%d", &number);

while (status != EOF)
{
   if (number < 0)
   {
      sum = sum + number;
   }
   status = fscanf(in, "%d", &number);
}

printf("The sum is %d.\n", sum);
fclose(in);
```

```
int number, sum = 0, status;
FILE* in;

in  = fopen("input.txt", "r");
status = fscanf(in, "%d", &number);

while (status != EOF)
{
    if (number < 0)
    {
        sum = sum + number;
    }
    status = fscanf(in, "%d", &number);
}

printf("The sum is %d.\n", sum);
fclose(in);
```

```
int number, sum = 0, status;
FILE* in;

in  = fopen("input.txt", "r");
status = fscanf(in, "%d", &number);

while (status != EOF)
{
   if (number < 0)
   {
      sum = sum + number;
   }
   status = fscanf(in, "%d", &number);
}

printf("The sum is %d.\n", sum);
fclose(in);
```

Solution

```
int number, sum = 0, status;
FILE* in;

in  = fopen("input.txt", "r");
status = fscanf(in, "%d", &number);

while (status != EOF)
{
   if (number < 0)
   {
      sum = sum + number;
   }
   status = fscanf(in, "%d", &number);
}

printf("The sum is %d.\n", sum);
fclose(in);
```

Write a code snippet that reads 11 integers from a file called input.txt and adds up all the negative numbers only.

Assume there are at least 11 numbers in the file.

Solution

```
int number, sum, status;
FILE* in;

in  = fopen("input.txt", "r");
sum = 0;

for (i = 0; i < 11; i++)
{
   fscanf(in, "%d", &number);

   if (number < 0)
      sum = sum + number;
}

printf("The sum is %d.\n", sum);
fclose(in);
```

```
int number, sum, status;
FILE* in;

in  = fopen("input.txt", "r");
sum = 0;

for (i = 0; i < 11; i++)
{
   fscanf(in, "%d", &number);

   if (number < 0)
      sum = sum + number;
}

printf("The sum is %d.\n", sum);
fclose(in);
```

```
int number, sum, status;
FILE* in;

in  = fopen("input.txt", "r");
sum = 0;

for (i = 0; i < 11; i++)
{
    fscanf(in, "%d", &number);

    if (number < 0)
       sum = sum + number;
}

printf("The sum is %d.\n", sum);
fclose(in);
```

```
int number, sum, status;
FILE* in;

in  = fopen("input.txt", "r");
sum = 0;

for (i = 0; i < 11; i++)
{
   fscanf(in, "%d", &number);

   if (number < 0)
      sum = sum + number;
}

printf("The sum is %d.\n", sum);
fclose(in);
```

Solution

```
int number, sum, status;
FILE* in;

in  = fopen("input.txt", "r");
sum = 0;

for (i = 0; i < 11; i++)
{
   fscanf(in, "%d", &number);

   if (number < 0)
      sum = sum + number;
}

printf("The sum is %d.\n", sum);
fclose(in);
```

A microwave oven manufacturer recommends that when heating two items, add 50% to the heating time, and when heating three items double the heating time. Heating more than three items is not recommended.

Write a code snippet that asks the user for the number of items and the sum of the single-item heating times. The program then writes out the recommended heating time.

Solution

```
int num_items, heating_time;

printf("Enter the number of items ");
printf("and the heating time in seconds: \n");

scanf("%d %d", &num_items, &heating_time);

if (num_items == 1)   /* one items */
   printf("Recommended heating time: %d\n",
      heating_time);
else if (num_items == 2) /* two items */
   printf("Recommended heating time: %d\n",
      (int)(heating_time + heating_time*0.5) );
else if (num_items == 3) /* three items */
   printf("Recommended heating time: %d\n",
      2*heating_time);
else
   printf("Not recommended\n");
```

```
int num_items, heating_time;

printf("Enter the number of items ");
printf("and the heating time in seconds: \n");

scanf("%d %d", &num_items, &heating_time);

if (num_items == 1)   /* one items */
   printf("Recommended heating time: %d\n",
      heating_time);
else if (num_items == 2) /* two items */
   printf("Recommended heating time: %d\n",
      (int)(heating_time + heating_time*0.5) );
else if (num_items == 3) /* three items */
   printf("Recommended heating time: %d\n",
      2*heating_time);
else
   printf("Not recommended\n");
```

```
int num_items, heating_time;

printf("Enter the number of items ");
printf("and the heating time in seconds: \n");

scanf("%d %d", &num_items, &heating_time);

if (num_items == 1)   /* one items */
   printf("Recommended heating time: %d\n",
      heating_time);
else if (num_items == 2) /* two items */
   printf("Recommended heating time: %d\n",
      (int)(heating_time + heating_time*0.5) );
else if (num_items == 3) /* three items */
   printf("Recommended heating time: %d\n",
      2*heating_time);
else
   printf("Not recommended\n");
```

```
int num_items, heating_time;

printf("Enter the number of items ");
printf("and the heating time in seconds: \n");

scanf("%d %d", &num_items, &heating_time);

if (num_items == 1)   /* one items */
   printf("Recommended heating time: %d\n",
      heating_time);
else if (num_items == 2) /* two items */
   printf("Recommended heating time: %d\n",
      (int)(heating_time + heating_time*0.5) );
else if (num_items == 3) /* three items */
   printf("Recommended heating time: %d\n",
      2*heating_time);
else
   printf("Not recommended\n");
```

```
int num_items, heating_time;

printf("Enter the number of items ");
printf("and the heating time in seconds: \n");

scanf("%d %d", &num_items, &heating_time);

if (num_items == 1)   /* one items */
   printf("Recommended heating time: %d\n",
      heating_time);
else if (num_items == 2) /* two items */
   printf("Recommended heating time: %d\n",
      (int)(heating_time + heating_time*0.5) );
else if (num_items == 3) /* three items */
   printf("Recommended heating time: %d\n",
      2*heating_time);
else
   printf("Not recommended\n");
```

```
int num_items, heating_time;

printf("Enter the number of items ");
printf("and the heating time in seconds: \n");

scanf("%d %d", &num_items, &heating_time);

if (num_items == 1)   /* one items */
   printf("Recommended heating time: %d\n",
      heating_time);
else if (num_items == 2) /* two items */
   printf("Recommended heating time: %d\n",
      (int)(heating_time + heating_time*0.5) );
else if (num_items == 3) /* three items */
   printf("Recommended heating time: %d\n",
      2*heating_time);
else
   printf("Not recommended\n");
```

```
int num_items, heating_time;

printf("Enter the number of items ");
printf("and the heating time in seconds: \n");

scanf("%d %d", &num_items, &heating_time);

if (num_items == 1)   /* one items */
   printf("Recommended heating time: %d\n",
      heating_time);
else if (num_items == 2) /* two items */
   printf("Recommended heating time: %d\n",
      (int)(heating_time + heating_time*0.5) );
else if (num_items == 3) /* three items */
   printf("Recommended heating time: %d\n",
      2*heating_time);
else
   printf("Not recommended\n");
```

```
int num_items, heating_time;

printf("Enter the number of items ");
printf("and the heating time in seconds: \n");

scanf("%d %d", &num_items, &heating_time);

if (num_items == 1)   /* one items */
   printf("Recommended heating time: %d\n",
      heating_time);
else if (num_items == 2) /* two items */
   printf("Recommended heating time: %d\n",
      (int)(heating_time + heating_time*0.5) );
else if (num_items == 3) /* three items */
   printf("Recommended heating time: %d\n",
      2*heating_time);
else
   printf("Not recommended\n");
```

```
int num_items, heating_time;

printf("Enter the number of items ");
printf("and the heating time in seconds: \n");

scanf("%d %d", &num_items, &heating_time);

if (num_items == 1)   /* one items */
   printf("Recommended heating time: %d\n",
      heating_time);
else if (num_items == 2) /* two items */
   printf("Recommended heating time: %d\n",
      (int)(heating_time + heating_time*0.5) );
else if (num_items == 3) /* three items */
   printf("Recommended heating time: %d\n",
      2*heating_time);
else
   printf("Not recommended\n");
```

Solution

```
int num_items, heating_time;

printf("Enter the number of items ");
printf("and the heating time in seconds: \n");

scanf("%d %d", &num_items, &heating_time);

if (num_items == 1)   /* one items */
   printf("Recommended heating time: %d\n",
      heating_time);
else if (num_items == 2) /* two items */
   printf("Recommended heating time: %d\n",
      (int)(heating_time + heating_time*0.5) );
else if (num_items == 3) /* three items */
   printf("Recommended heating time: %d\n",
      2*heating_time);
else
   printf("Not recommended\n");
```

```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```

```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```

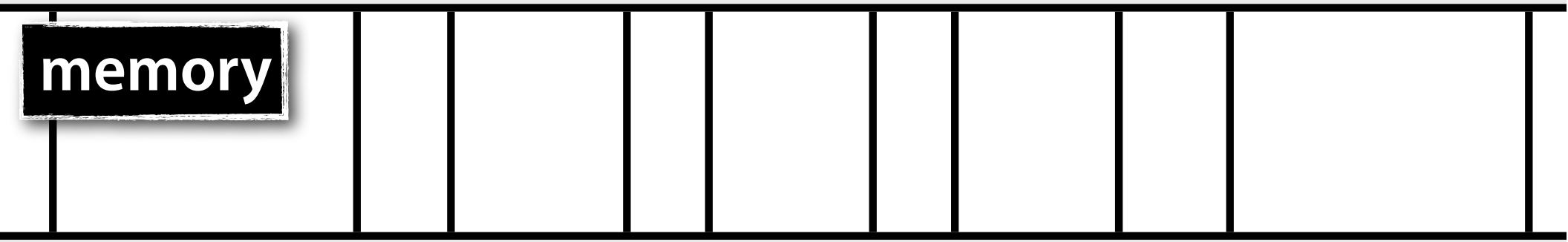
What's the output?

```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```

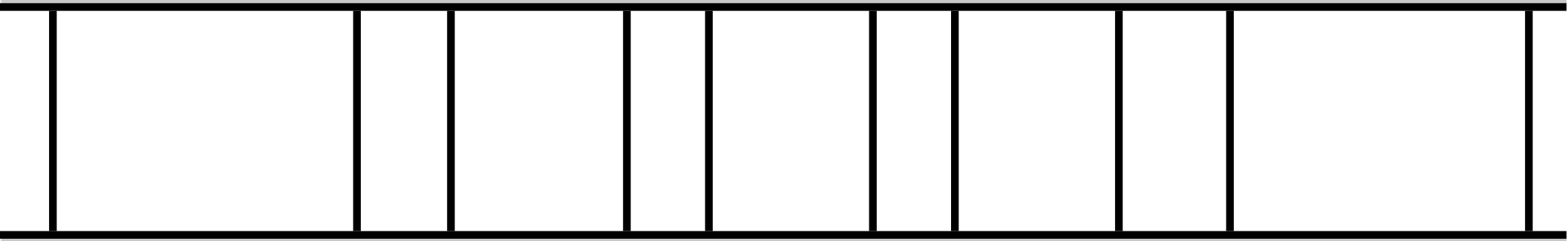
Output

```
i = 5, j = 10
```

```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```

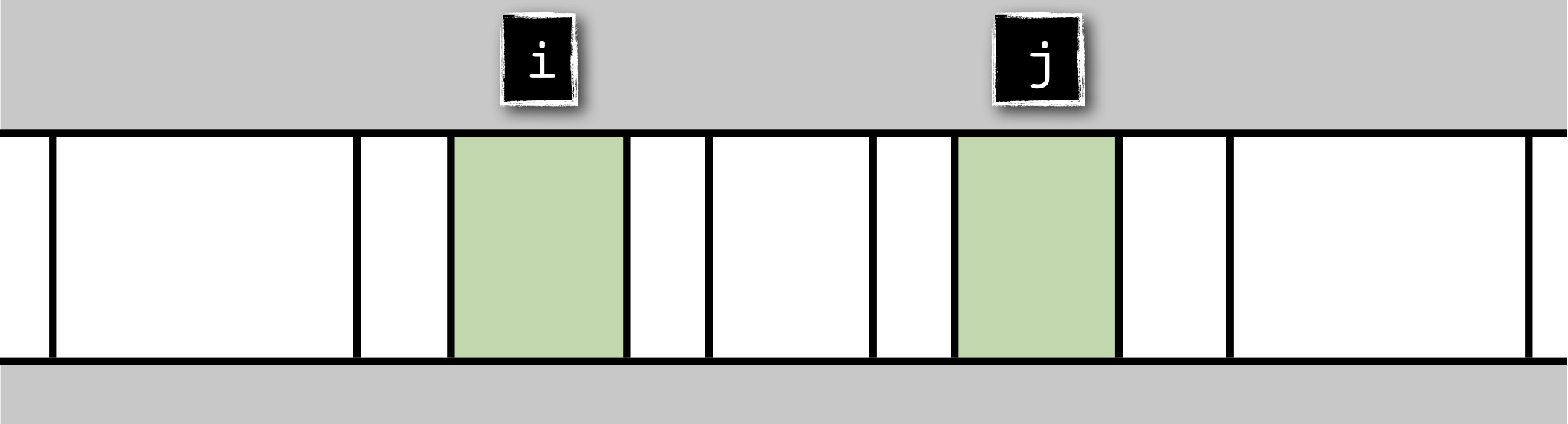


```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```

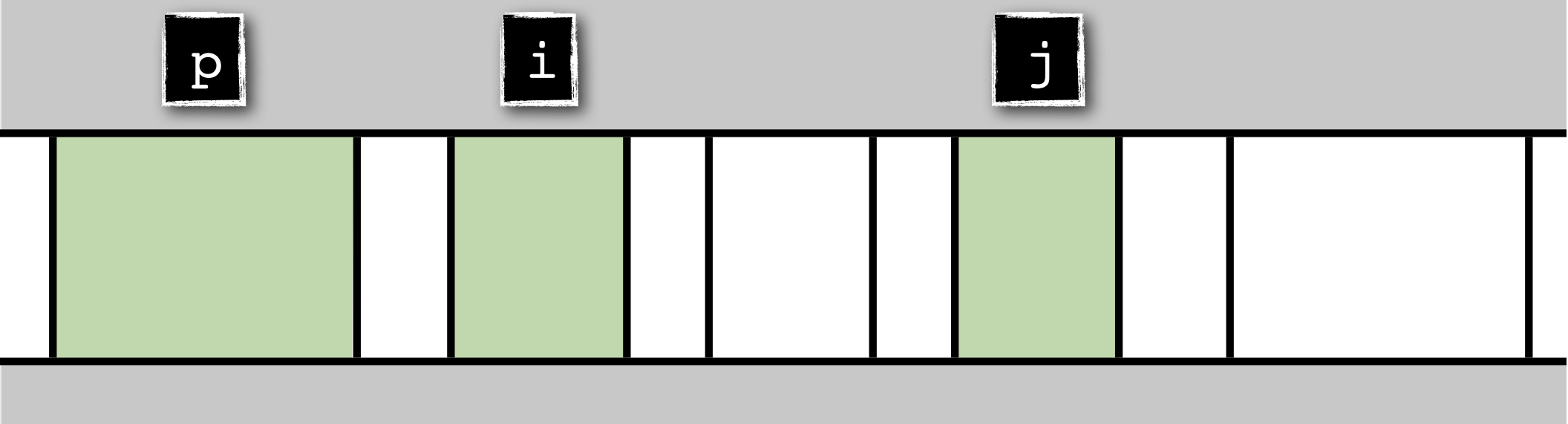


```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```

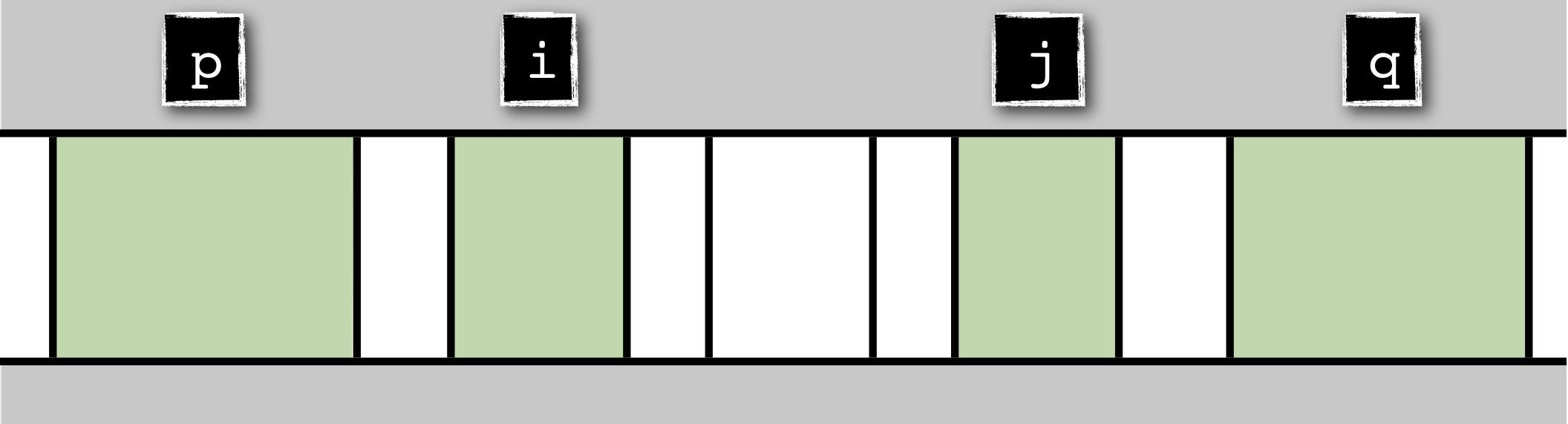
```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```



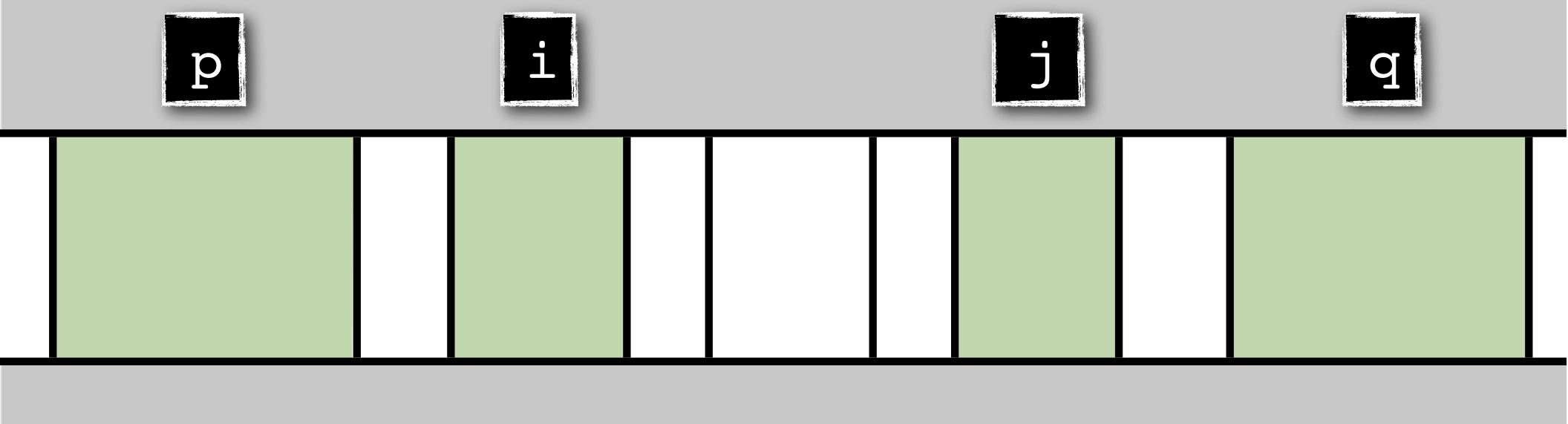
```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```



```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```



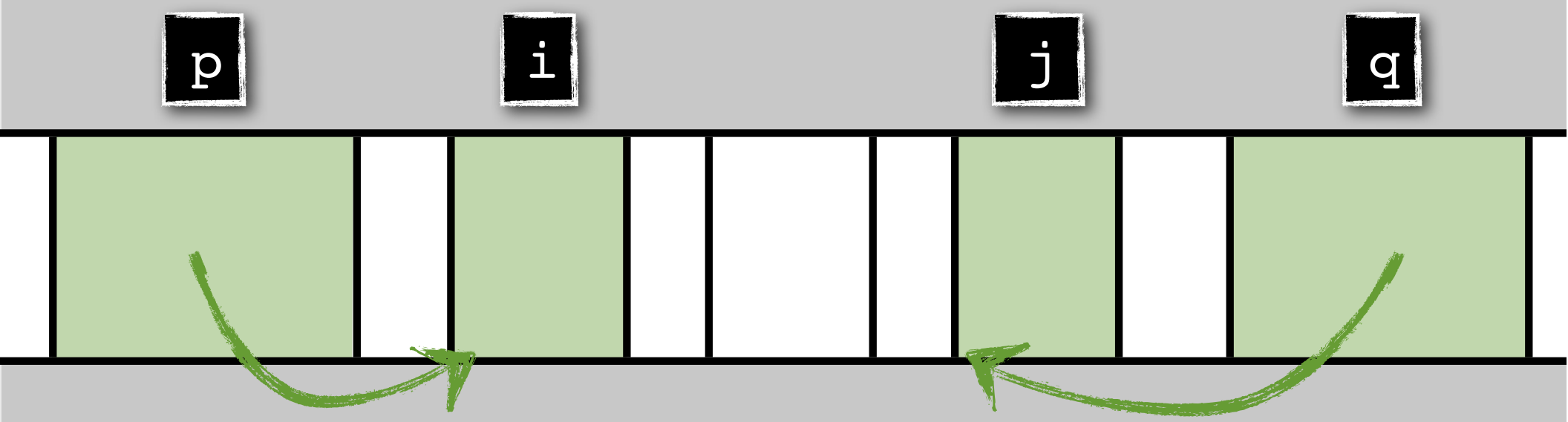
```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```



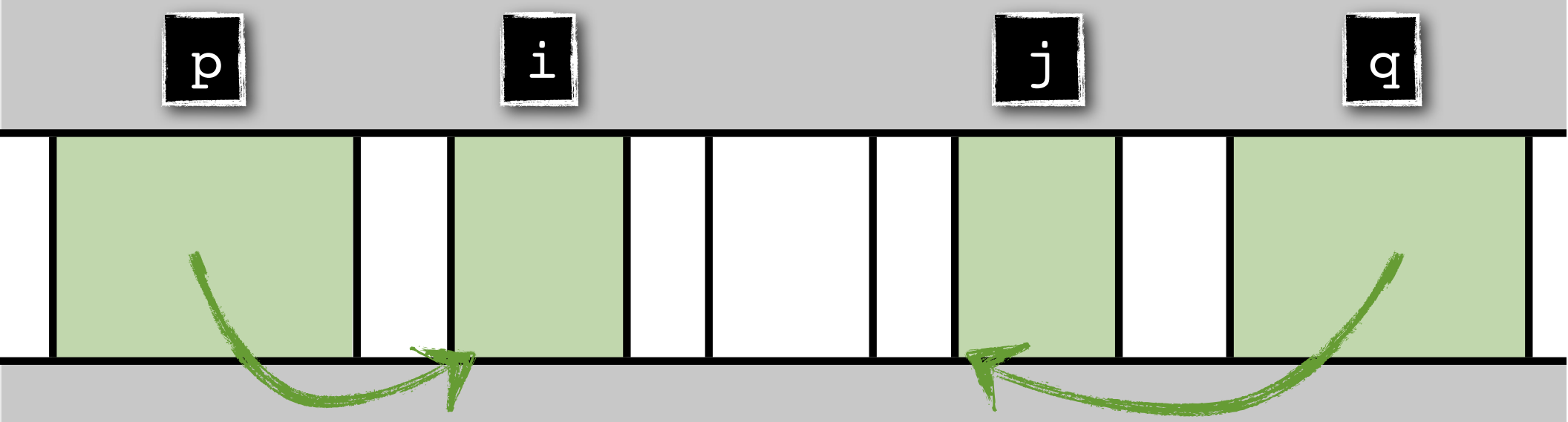
```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```

```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```

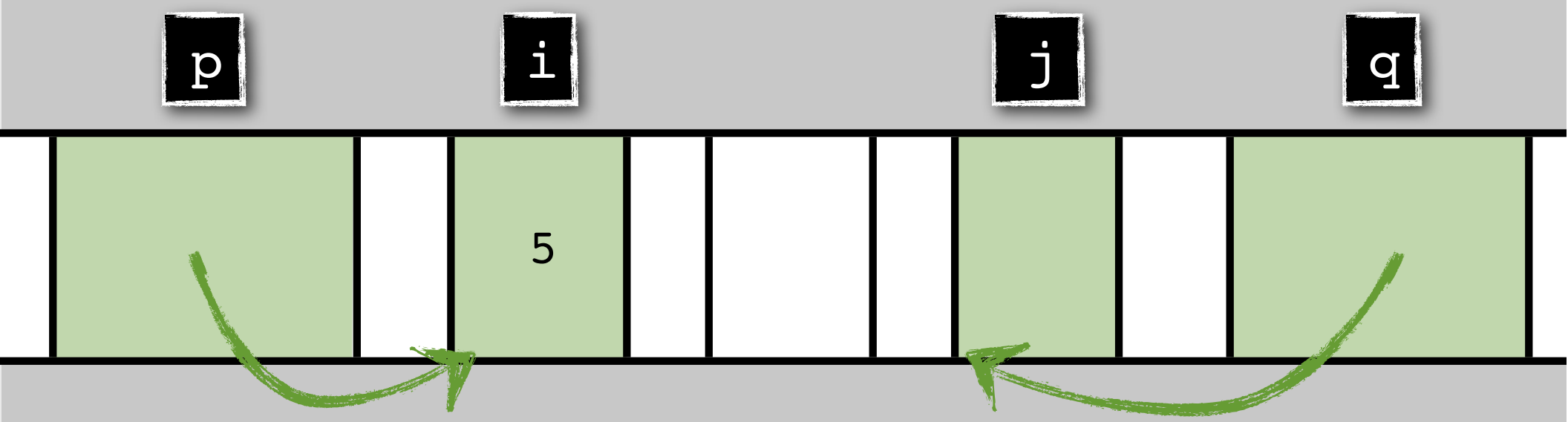
```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```



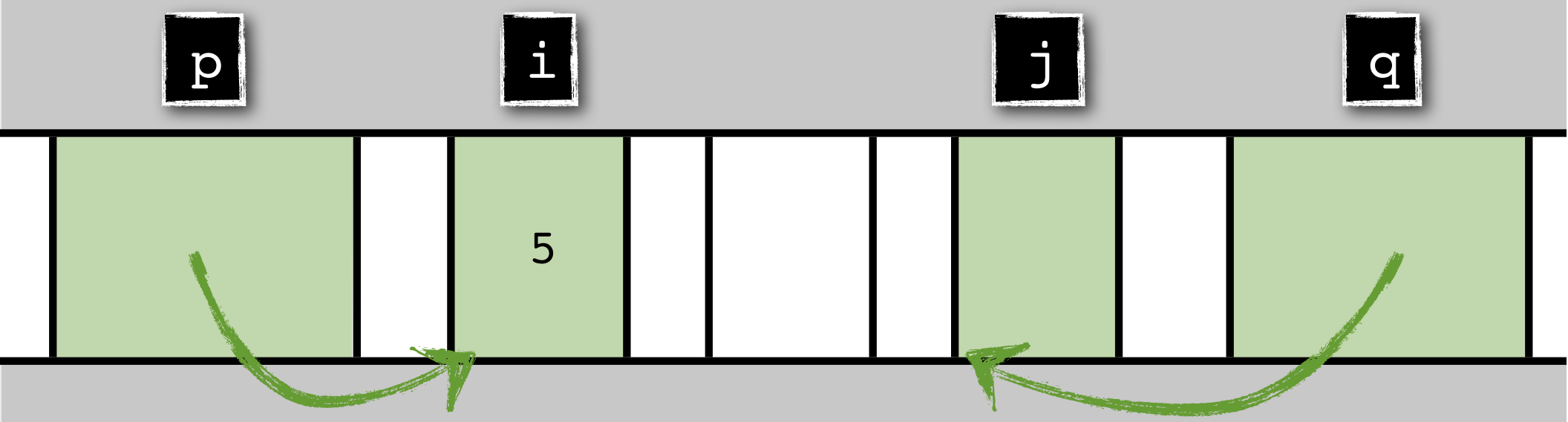
```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```



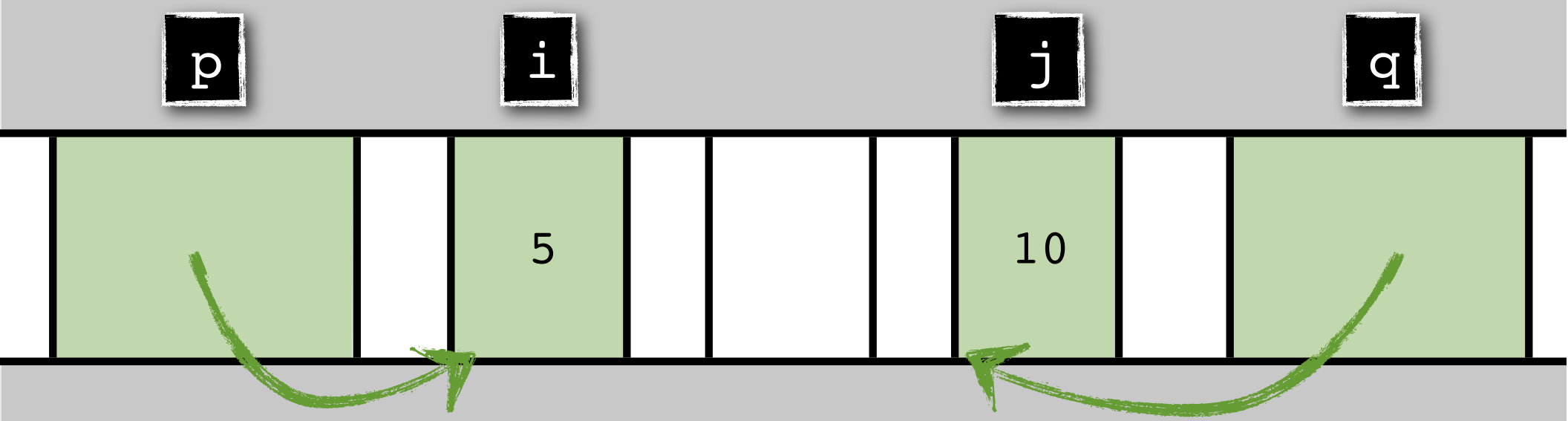
```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```



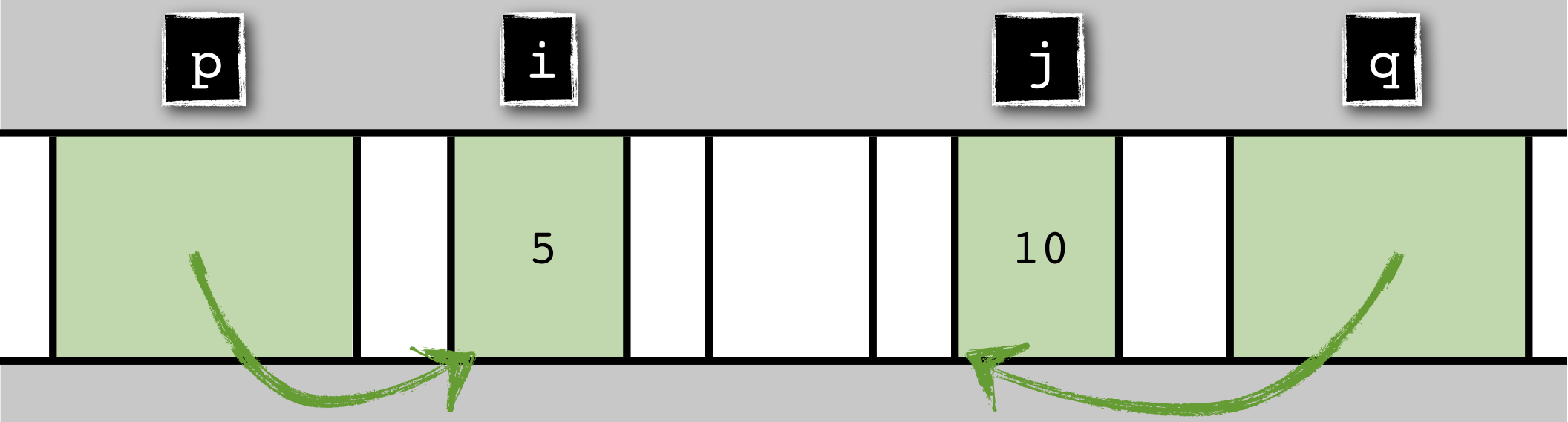
```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```



```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```



```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```



```
int i, j, *p, *q;
p = &i;
q = &j;
*p = 5;
*q = *p + i;

printf("i = %d, j = %d\n", i, j);
```