

CSE1301
Computer Programming
Lecture 10:
Iteration (Part 1)

1

- Topics**
- **while** statement
 - **for** statement
 - **break** statement
 - Nested loops
- 2

The while Statement

Implements the repetition in an algorithm

- Repeatedly executes a block of statements
- Tests a condition (Boolean expression) at the start of each iteration
- Terminates when condition becomes false (zero)

3

Example: addnum.c

```

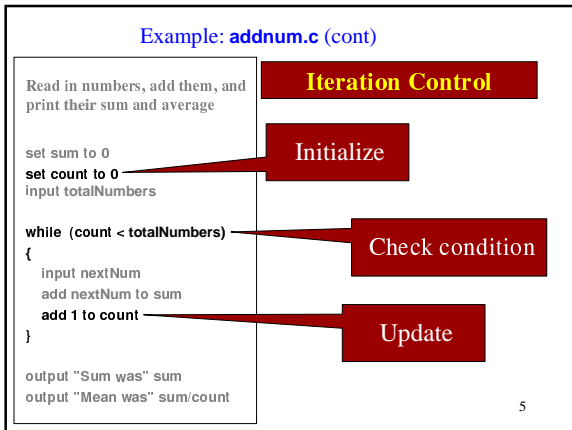
Read in numbers, add them, and
print their sum and average

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
  input nextNum
  add nextNum to sum
  add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
  
```

4



Example: addnum.c (cont)

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
  set sum to 0
  set count to 0
  input totalNumbers

  while (count < totalNumbers)
  {
    input nextNum
    add nextNum to sum
    add 1 to count
  }

  output "Sum was" sum
  output "Mean was" sum/count

  return 0;
}
  
```

6

Example: addnum.c (cont)

```

Read in numbers, add them, and
print their sum and average

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;

    only the variables sum and
    count are initialized to 0

    return 0;
}
    
```

7

Example: addnum.c (cont)

```

Read in numbers, add them, and
print their sum and average

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    return 0;
}
    
```

8

Example: addnum.c (cont)

```

Read in numbers, add them, and
print their sum and average

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    while (count < totalNumbers)
    {
        input nextNum
        add nextNum to sum
        add 1 to count
    }

    return 0;
}
    
```

9

Example: addnum.c (cont)

```

Read in numbers, add them, and
print their sum and average

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    while (count < totalNumbers)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
        count++;
    }

    return 0;
}
    
```

10

Example: addnum.c (cont)

```

Read in numbers, add them, and
print their sum and average

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    while (count < totalNumbers)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
        count++;
    }

    return 0;
}
    
```

Same as: **sum = sum + nextNum;**
 Others: **-=, *=, /=, etc. (King, Table 4.2)**

Example: addnum.c (cont)

```

Read in numbers, add them, and
print their sum and average

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    while (count < totalNumbers)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
        count++;
    }

    return 0;
}
    
```

Same as: **count = count + 1;**
 Decrement: **count --;** (King, Table 4.2)

Example: addnum.c (cont)

Read in numbers, add them, and print their sum and average

```

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    while (count < totalNumbers)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
        count++;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);
    return 0;
}
    
```

13

Example: addnum.c (cont)

Read in numbers, add them, and print their sum and average

```

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    while (count < totalNumbers)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
        count++;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);
    return 0;
}
    
```

14

Example: addnum.c (cont)

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    while (count < totalNumbers)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
        count++;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);
    return 0;
}
    
```

totalNumbers	count	nextNum	sum
???	0	???	0.0

15

Example: addnum.c (cont)

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    while (count < totalNumbers)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
        count++;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);
    return 0;
}
    
```

totalNumbers	count	nextNum	sum
???	0	???	0.0
3			

16

Example: addnum.c (cont)

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    while (count < totalNumbers)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
        count++;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);
    return 0;
}
    
```

totalNumbers	count	nextNum	sum
???	0	???	0.0
3			
	1	4	4.0

17

Example: addnum.c (cont)

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    while (count < totalNumbers)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
        count++;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);
    return 0;
}
    
```

totalNumbers	count	nextNum	sum
???	0	???	0.0
3			
	1	4	4.0
	2	-1	3.0

18

```
#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count = 0, totalNumbers;
    scanf("%d", &totalNumbers);

    while (count < totalNumbers)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
        count++;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);
    return 0;
}
```

Example: **addnum.c** (cont)

totalNumbers	count	nextNum	sum
????	0	????	0.0
3			
	1	4	4.0
	2	-1	3.0
	3	6.2	9.2

Common Mistakes in **while** -- "one liners"

```
while (num < minimum)
    scanf("%d", &num);
    printf("Number must be greater than %d.\n", minimum);
    printf("Please try again.\n");
```

```
while (num < minimum)
{
    scanf("%d", &num);
}

printf("Number must be greater than %d.\n", minimum);
printf("Please try again.\n");
```

Common Mistakes in **while** -- "one liners" (cont)

```
while (num < minimum)
    scanf("%d", &num);
    printf("Number must be greater than %d.\n", minimum);
    printf("Please try again.\n");
```

```
while (num < minimum)
{
    scanf("%d", &num);
    printf("Number must be greater than %d.\n", minimum);
    printf("Please try again.\n");
}
```

✔

Common Mistakes in **while** -- extra semi-colon;

```
while (num < minimum);
{
    scanf("%d", &num);
    printf("Number must be greater than %d.\n", minimum);
    printf("Please try again.\n");
}
```

Marks the end of the while-block -- usual cause of infinite loops

Checking for End-of-Input / End-of-File in **while**

Read in numbers, add them, and print their sum and average

set sum to 0

input nextNum

check if end of input

while (not end of input)

```
{
    add nextNum to sum
    input nextNum
    check if end of input
}
```

etc...etc...etc...

Checking for End-of-Input / End-of-File in **while** (cont)

Read in numbers, add them, and print their sum and average

set sum to 0

input nextNum

check if end of input

while (not end of input)

```
{
    add nextNum to sum
    input nextNum
    check if end of input
}
```

etc...etc...etc...

etc...etc...etc...

```
float nextNum;
float sum = 0.0;

scanf("%f", &nextNum);

while ( ?????? )
{
    sum += nextNum;
    scanf("%f", &nextNum);
}

etc...etc...etc...
```

Checking for End-of-Input / End-of-File in **while** (cont)

<p>Read in numbers, add them, and print their sum and average</p> <pre> set sum to 0 input nextNum check if end of input while (not end of input) { add nextNum to sum input nextNum check if end of input } </pre>	<p><i>etc...etc...etc...</i></p> <pre> float nextNum; float sum = 0.0; scanf("%f", &nextNum); ??????? while (??????) { sum += nextNum; scanf("%f", &nextNum); ??????? } </pre> <p><i>etc...etc...etc...</i></p>
--	--

25

Checking for End-of-Input / End-of-File in **while** (cont)

<p>Read in numbers, add them, and print their sum and average</p> <pre> set sum to 0 input nextNum check if end of input while (not end of input) { add nextNum to sum input nextNum check if end of input } </pre>	<p><i>etc...etc...etc...</i></p> <pre> float nextNum; float sum = 0.0; scanf("%f", &nextNum); ??????? while (??????) { sum += nextNum; scanf("%f", &nextNum); ??????? } </pre> <p><i>etc...etc...etc...</i></p>
--	--

Recall: When the input ends, the scanf() function returns a special char value: EOF

26

Checking for End-of-Input / End-of-File in **while** (cont)

<p>Read in numbers, add them, and print their sum and average</p> <pre> set sum to 0 input nextNum check if end of input while (not end of input) { add nextNum to sum input nextNum check if end of input } </pre>	<p><i>etc...etc...etc...</i></p> <pre> float nextNum; float sum = 0.0; while (scanf("%f",&nextNum) != EOF) { sum += nextNum; } </pre> <p><i>etc...etc...etc...</i></p>
--	---

27

Checking for End-of-Input / End-of-File in **while** (cont)

<p>Read in numbers, add them, and print their sum and average</p> <pre> set sum to 0 input nextNum check if end of input while (not end of input) { add nextNum to sum input nextNum check if end of input } </pre>	<p><i>etc...etc...etc...</i></p> <pre> float nextNum; float sum = 0.0; while (scanf("%f",&nextNum) != EOF) { sum += nextNum; } </pre> <p><i>etc...etc...etc...</i></p>
--	---

28

The **for** Statement

- Form of loop which allows for initialization and iteration control
- Syntax:

```

for ( initialization; condition; update )
{
  instruction block
}
                
```

Careful! A semi-colon here marks the end of the instruction block!

29

Example: addfor.c

```

Read in numbers, add them, and print the sum and the average

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
  input nextNum
  add nextNum to sum
  add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
                
```

30

Example: addfor.c (cont)

Read in numbers, add them, and print the sum and the average

```

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count, totalNumbers;

    scanf("%d", &totalNumbers);

    for (count=0;
         count < totalNumbers;
         count++)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);

    return 0;
}
    
```

31

Example: addfor.c (cont)

Read in numbers, add them, and print the sum and the average

```

set sum to 0
set count to 0
input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum;
    int count, totalNumbers;

    scanf("%d", &totalNumbers);

    for (count=0;
         count < totalNumbers;
         count++)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);

    return 0;
}
    
```

Initialize

32

Example: addfor.c (cont)

Read in numbers, add them, and print the sum and the average

```

set sum to 0
set count to 0

input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum;
    int count, totalNumbers;

    scanf("%d", &totalNumbers);

    for (count=0;
         count < totalNumbers;
         count++)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);

    return 0;
}
    
```

Check condition

33

Example: addfor.c (cont)

Read in numbers, add them, and print the sum and the average

```

set sum to 0
set count to 0

input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum;
    int count, totalNumbers;

    scanf("%d", &totalNumbers);

    for (count=0;
         count < totalNumbers;
         count++)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);

    return 0;
}
    
```

Update (aka Increment Step)

IMPORTANT!!
The Update is performed AFTER the body of the loop

34

Example: addfor.c (cont)

Read in numbers, add them, and print the sum and the average

```

set sum to 0
set count to 0

input totalNumbers

while (count < totalNumbers)
{
    input nextNum
    add nextNum to sum
    add 1 to count
}

output "Sum was" sum
output "Mean was" sum/count
    
```

```

#include <stdio.h>
/*****
Read in numbers and add them up
Print out the sum and the average
*****/
int main()
{
    float nextNum, sum = 0.0;
    int count, totalNumbers;

    scanf("%d", &totalNumbers);

    for (count=0;
         count < totalNumbers;
         count++)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",sum/count);

    return 0;
}
    
```

35

while and for

```

#include <stdio.h>
int main()
{
    float nextNum, sum = 0.0;
    int count, totalNumbers;

    scanf("%d", &totalNumbers);

    count = 0;
    while (count < totalNumbers)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
        count++;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",
           sum/count);

    return 0;
}
    
```

```

#include <stdio.h>
int main()
{
    float nextNum, sum = 0.0;
    int count, totalNumbers;

    scanf("%d", &totalNumbers);

    for (count=0;
         count < totalNumbers;
         count++)
    {
        scanf("%f", &nextNum);
        sum += nextNum;
    }

    printf("Sum was %f\n",sum);
    printf("Mean was %f\n",
           sum/count);

    return 0;
}
    
```

36

while and for (cont)

<pre>#include <stdio.h> int main() { float nextNum, sum = 0.0; int count, totalNumbers; scanf("%d", &totalNumbers); count = 0; while (count < totalNumbers) { scanf("%f", &nextNum); sum += nextNum; count++; } printf("Sum was %f\n", sum); printf("Mean was %f\n", sum/count); return 0; }</pre>	<pre>#include <stdio.h> int main() { float nextNum, sum = 0.0; int count, totalNumbers; scanf("%d", &totalNumbers); for (count=0; count < totalNumbers; count++) { scanf("%f", &nextNum); sum += nextNum; } printf("Sum was %f\n", sum); printf("Mean was %f\n", sum/count); return 0; }</pre>
---	--

Initialize

37

while and for (cont)

<pre>#include <stdio.h> int main() { float nextNum, sum = 0.0; int count, totalNumbers; scanf("%d", &totalNumbers); count = 0; while (count < totalNumbers) { scanf("%f", &nextNum); sum += nextNum; count++; } printf("Sum was %f\n", sum); printf("Mean was %f\n", sum/count); return 0; }</pre>	<pre>#include <stdio.h> int main() { float nextNum, sum = 0.0; int count, totalNumbers; scanf("%d", &totalNumbers); for (count=0; count < totalNumbers; count++) { scanf("%f", &nextNum); sum += nextNum; } printf("Sum was %f\n", sum); printf("Mean was %f\n", sum/count); return 0; }</pre>
--	--

Check condition

38

while and for (cont)

<pre>#include <stdio.h> int main() { float nextNum, sum = 0.0; int count, totalNumbers; scanf("%d", &totalNumbers); count = 0; while (count < totalNumbers) { scanf("%f", &nextNum); sum += nextNum; count++; } printf("Sum was %f\n", sum); printf("Mean was %f\n", sum/count); return 0; }</pre>	<pre>#include <stdio.h> int main() { float nextNum, sum = 0.0; int count, totalNumbers; scanf("%d", &totalNumbers); for (count=0; count < totalNumbers; count++) { scanf("%f", &nextNum); sum += nextNum; } printf("Sum was %f\n", sum); printf("Mean was %f\n", sum/count); return 0; }</pre>
--	--

Update

39

The break Statement

- Implements the "exit loop" primitive
- Causes flow of control to leave a loop block (**while** or **for**) immediately

40

Example: recip.c

Print out the reciprocals of numbers entered. Quit when 0 is entered

```
loop
{
    input nextNum
    if (nextNum is 0)
    {
        exit loop
    }
    else
    {
        output 1/nextNum
    }
}
```

41

Example: recip.c (cont)

```
#include <stdio.h>
/*****\
Print out the reciprocals of
numbers entered. Quit when 0
is entered
\*****/

int main()
{
    float nextNum;

loop
{
    input nextNum
    if (nextNum is 0)
    {
        exit loop
    }
    else
    {
        output 1/nextNum
    }
}

return 0;
}
```

42

Example: recip.c (cont)

Print out the reciprocals of numbers entered. Quit when 0 is entered

```

loop
{
  input nextNum
  if (nextNum is 0)
  {
    exit loop
  }
  else
  {
    output 1/nextNum
  }
}
    
```

```

#include <stdio.h>
/*****
Print out the reciprocals of
numbers entered. Quit when 0
is entered
*****/

int main()
{
  float nextNum;

  while (1)
  {
    input nextNum
    if (nextNum is 0)
    {
      exit loop
    }
    else
    {
      output 1/nextNum
    }
  }
  return 0;
}
    
```

“while (True)” infinite loop

43

Example: recip.c (cont)

Print out the reciprocals of numbers entered. Quit when 0 is entered

```

loop
{
  input nextNum
  if (nextNum is 0)
  {
    exit loop
  }
  else
  {
    output 1/nextNum
  }
}
    
```

```

#include <stdio.h>
/*****
Print out the reciprocals of
numbers entered. Quit when 0
is entered
*****/

int main()
{
  float nextNum;

  while (1)
  {
    scanf("%f", &nextNum);
  }
  return 0;
}
    
```

44

Example: recip.c (cont)

Print out the reciprocals of numbers entered. Quit when 0 is entered

```

loop
{
  input nextNum
  if (nextNum is 0)
  {
    exit loop
  }
  else
  {
    output 1/nextNum
  }
}
    
```

```

#include <stdio.h>
/*****
Print out the reciprocals of
numbers entered. Quit when 0
is entered
*****/

int main()
{
  float nextNum;

  while (1)
  {
    scanf("%f", &nextNum);
    if (nextNum == 0.0)
    {
      break;
    }
    else
    {
      printf("%f\n", 1/nextNum);
    }
  }
  return 0;
}
    
```

45

Example: recip.c (cont)

Print out the reciprocals of numbers entered. Quit when 0 is entered

```

loop
{
  input nextNum
  if (nextNum is 0)
  {
    exit loop
  }
  else
  {
    output 1/nextNum
  }
}
    
```

```

#include <stdio.h>
/*****
Print out the reciprocals of
numbers entered. Quit when 0
is entered
*****/

int main()
{
  float nextNum;

  while (1)
  {
    scanf("%f", &nextNum);
    if (nextNum==0.0)
    {
      break;
    }
    else
    {
      printf("%f\n", 1/nextNum);
    }
  }
  return 0;
}
    
```

46

Example: recip.c (cont)

Print out the reciprocals of numbers entered. Quit when 0 is entered

```

loop
{
  input nextNum
  if (nextNum is 0)
  {
    exit loop
  }
  else
  {
    output 1/nextNum
  }
}
    
```

```

#include <stdio.h>
/*****
Print out the reciprocals of
numbers entered. Quit when 0
is entered
*****/

int main()
{
  float nextNum;

  while (1)
  {
    scanf("%f", &nextNum);
    if (nextNum==0.0)
    {
      break;
    }
    else
    {
      printf("%f\n", 1/nextNum);
    }
  }
  return 0;
}
    
```

47

Example: addpos.c

Read in numbers, and add only the positive ones. Quit when input is 0

```

set sum to 0

loop
{
  input number
  if (number is zero)
  {
    exit loop
  }
  else if ( number is positive)
  {
    add number to sum
  }
}

output sum
    
```

48

Example: addpos.c (cont)

```

Read in numbers, and add
only the positive ones. Quit
when input is 0

set sum to 0

loop
{
  input number
  if (number is zero)
  {
    exit loop
  }
  else if ( number is positive)
  {
    add number to sum
  }
}

output sum
    
```

```

include <stdio.h>

/*****
** Read in numbers, and add
** only the positive ones.
** Quit when input is 0
*****/

int main()
{
  float num, sum = 0.0;

  printf("sum = %f\n", sum);
  return 0;
}
    
```

49

Example: addpos.c (cont)

scanf returns EOF if an end of file occurs; otherwise it returns the number of items converted and assigned

```

Read in numbers, and add
only the positive ones. Quit
when input is 0

set sum to 0

loop
{
  input number
  if (number is zero)
  {
    exit loop
  }
  else if ( number is positive)
  {
    add number to sum
  }
}

output sum
    
```

```

include <stdio.h>

/*****
** Read in numbers, and add
** only the positive ones.
** Quit when input is 0
*****/

int main()
{
  float num, sum = 0.0;

  while (scanf("%f", &num) > 0)
  {
    if (num == 0)
      break;
    else if (num > 0)
      sum += num;
  }

  printf("sum = %f\n", sum);
  return 0;
}
    
```

50

Example: addpos.c (cont)

```

Read in numbers, and add
only the positive ones. Quit
when input is 0

set sum to 0

loop
{
  input number
  if (number is zero)
  {
    exit loop
  }
  else if ( number is positive)
  {
    add number to sum
  }
}

output sum
    
```

```

include <stdio.h>

/*****
** Read in numbers, and add
** only the positive ones.
** Quit when input is 0
*****/

int main()
{
  float num, sum = 0.0;

  while (scanf("%f", &num) > 0)
  {
    if (num == 0)
      break;
    else if (num > 0)
      sum += num;
  }

  printf("sum = %f\n", sum);
  return 0;
}
    
```

51

Example: addpos.c (cont)

```

Read in numbers, and add
only the positive ones. Quit
when input is 0

set sum to 0

loop
{
  input number
  if (number is zero)
  {
    exit loop
  }
  else if ( number is positive)
  {
    add number to sum
  }
}

output sum
    
```

```

include <stdio.h>

/*****
** Read in numbers, and add
** only the positive ones.
** Quit when input is 0
*****/

int main()
{
  float num, sum = 0.0;

  while (scanf("%f", &num) > 0)
  {
    if (num == 0)
      break;
    else if (num > 0)
      sum += num;
  }

  printf("sum = %f\n", sum);
  return 0;
}
    
```

52

Example: addpos.c (cont)

These comparisons are ok despite num being of type float

```

include <stdio.h>

/*****
** Read in numbers, and add
** only the positive ones.
** Quit when input is 0
*****/

int main()
{
  float num, sum = 0.0;

  while (scanf("%f", &num) > 0)
  {
    if (num == 0)
      break;
    else if (num > 0)
      sum += num;
  }

  printf("sum = %f\n", sum);
  return 0;
}
    
```

53

Example: addpos.c (cont)

```

Read in numbers, and add
only the positive ones. Quit
when input is 0

set sum to 0

loop
{
  input number
  if (number is zero)
  {
    exit loop
  }
  else if ( number is positive)
  {
    add number to sum
  }
}

output sum
    
```

```

include <stdio.h>

/*****
** Read in numbers, and add
** only the positive ones.
** Quit when input is 0
*****/

int main()
{
  float num, sum = 0.0;

  while (scanf("%f", &num) > 0)
  {
    if (num == 0)
      break;
    else if (num > 0)
      sum += num;
  }

  printf("sum = %f\n", sum);
  return 0;
}
    
```

54

scanf and while -- Example 1

```
float num;
while (scanf("%f", &num) > 0)
{
    ...etc...etc...etc...
}
```

Input: 45.2

Result: 1

55

scanf and while -- Example 1 (cont)

```
float num;
while (scanf("%f", &num) > 0)
{
    ...etc...etc...etc...
}
```

Input: -5

Result: 1

56

scanf and while -- Example 1 (cont)

```
float num;
while (scanf("%f", &num) > 0)
{
    ...etc...etc...etc...
}
```

Input: 0

Result: 1

57

scanf and while -- Example 1 (cont)

```
float num;
while (scanf("%f", &num) > 0)
{
    ...etc...etc...etc...
}
```

Input: c

Result: 0

58

scanf and while -- Example 1 (cont)

```
float num;
while (scanf("%f", &num) > 0)
{
    ...etc...etc...etc...
}
```

Input: Dog

Result: 0

59

scanf and while -- Example 1 (cont)

```
float num;
while (scanf("%f", &num) > 0)
{
    ...etc...etc...etc...
}
```

Input: ^Z or ^D
(depending on the operating system)

Result: EOF
(usually has value -1, but it can be any negative number)

60

scanf -- Example 2

```
int val;
float x, y, z;

val = scanf("%f %f %f", &x, &y, &z);
printf("%d\n", val);
```

Input: 42.5 -5 23
Output: 3

61

scanf -- Example 2 (cont)

```
int val;
float x, y, z;

val = scanf("%f %f %f", &x, &y, &z);
printf("%d\n", val);
```

Input: 42.5 -5 c
Output: 2

62

scanf -- Example 2 (cont)

```
int val;
float x, y, z;

val = scanf("%f %f %f", &x, &y, &z);
printf("%d\n", val);
```

Input: 42.5 c 23
Output: 1

63

scanf -- Example 2 (cont)

```
int val;
float x, y, z;

val = scanf("%f %f %f", &x, &y, &z);
printf("%d\n", val);
```

Input: man 2 wolf
Output: 0

64

scanf -- Example 3

```
int num;
char ch;
float x;

printf("Please enter an int, a char, and a float: ");
if ( scanf("%d %c %f", &num, &ch, &x) != 3 )
{
    printf("Invalid input. No cookie for you.\n");
}
else
{
    printf("Thank you. Your cookie is in the box.\n");
}
```

65

scanf -- Example 3 (cont)

```
int num;
char ch;
float x;

printf("Please enter an int, a char, and a float: ");
if ( scanf("%d %c %f", &num, &ch, &x) != 3 )
{
    printf("Invalid input. No cookie for you.\n");
}
else
{
    printf("Thank you. Your cookie is in the box.\n");
}
```

66

Nested Loops

- Loops can be placed inside other loops
- The **break** statement applies to the innermost enclosing **while** or **for** statement

67

Example: rect.c

Print an m by n rectangle of asterisks

input width and height

```
for each row
{
  for each column in the current row
  {
    print an asterisk
  }
  start next row
}
```

68

Example: rect.c (cont)

Print an m by n rectangle of asterisks

input width and height

```
for each row
{
  for each column in the current row
  {
    print an asterisk
  }
  start next row
}
```

```
#include <stdio.h>
/* Print an m-by-n rectangle of asterisks */
int main()
{
  int rowc, colc, numrow, numcol;

  printf("\nEnter width: ");
  scanf("%d", &numcol);
  printf("\nEnter height: ");
  scanf("%d", &numrow);

  return 0;
}
```

69

Example: rect.c (cont)

Print an m by n rectangle of asterisks

input width and height

```
for each row
{
  for each column in the current row
  {
    print an asterisk
  }
  start next row
}
```

```
#include <stdio.h>
/* Print an m-by-n rectangle of asterisks */
int main()
{
  int rowc, colc, numrow, numcol;

  printf("\nEnter width: ");
  scanf("%d", &numcol);
  printf("\nEnter height: ");
  scanf("%d", &numrow);

  for (rowc=0; rowc < numrow; rowc++)
  {
    for (colc=0; colc < numcol; colc++)
    {
      printf("**");
    }
  }
  return 0;
}
```

70

Example: rect.c (cont)

Print an m by n rectangle of asterisks

input width and height

```
for each row
{
  for each column in the current row
  {
    print an asterisk
  }
  start next row
}
```

```
#include <stdio.h>
/* Print an m-by-n rectangle of asterisks */
int main()
{
  int rowc, colc, numrow, numcol;

  printf("\nEnter width: ");
  scanf("%d", &numcol);
  printf("\nEnter height: ");
  scanf("%d", &numrow);

  for (rowc=0; rowc < numrow; rowc++)
  {
    for (colc=0; colc < numcol; colc++)
    {
      printf("**");
    }
  }
  return 0;
}
```

71

Example: rect.c (cont)

Print an m by n rectangle of asterisks

input width and height

```
for each row
{
  for each column in the current row
  {
    print an asterisk
  }
  start next row
}
```

```
#include <stdio.h>
/* Print an m-by-n rectangle of asterisks */
int main()
{
  int rowc, colc, numrow, numcol;

  printf("\nEnter width: ");
  scanf("%d", &numcol);
  printf("\nEnter height: ");
  scanf("%d", &numrow);

  for (rowc=0; rowc < numrow; rowc++)
  {
    for (colc=0; colc < numcol; colc++)
    {
      printf("**");
    }
  }
  return 0;
}
```

72

Example: rect.c (cont)

Print an m by n rectangle of asterisks

input width and height

for each row

{

 for each column in the current row

 {

 print an asterisk

 }

 start next row

}

```
#include <stdio.h>
/* Print an m-by-n rectangle of asterisks */
int main()
{
    int rowc, colc, numrow, numcol;

    printf("\nEnter width: ");
    scanf("%d", &numcol);
    printf("\nEnter height: ");
    scanf("%d", &numrow);

    for (rowc=0; rowc < numrow; rowc++)
    {
        for (colc=0; colc < numcol; colc++)
        {
            printf("***");
        }
        printf("\n");
    }
    return 0;
}
```

73

Example: rect.c (cont)

Print an m by n rectangle of asterisks

input width and height

for each row

{

 for each column in the current row

 {

 print an asterisk

 }

 start next row

}

```
#include <stdio.h>
/* Print an m-by-n rectangle of asterisks */
int main()
{
    int rowc, colc, numrow, numcol;

    printf("\nEnter width: ");
    scanf("%d", &numcol);
    printf("\nEnter height: ");
    scanf("%d", &numrow);

    for (rowc=0; rowc < numrow; rowc++)
    {
        for (colc=0; colc < numcol; colc++)
        {
            printf("***");
        }
        printf("\n");
    }
    return 0;
}
```

74

Example: rect.c (cont)

Print an m by n rectangle of asterisks

input width and height

algorithm

for each row

{

 for each column in the current row

 {

 print an asterisk

 }

 start next row

}

program

```
#include <stdio.h>
/* Print an m-by-n rectangle of asterisks */
int main()
{
    int rowc, colc, numrow, numcol;

    printf("\nEnter width: ");
    scanf("%d", &numcol);
    printf("\nEnter height: ");
    scanf("%d", &numrow);

    for (rowc=0; rowc < numrow; rowc++)
    {
        for (colc=0; colc < numcol; colc++)
        {
            printf("***");
        }
        printf("\n");
    }
    return 0;
}
```

75

Variation: rect2.c

Print an m by n rectangle of asterisks

input width and height

for each row

{

 for each column in the current row

 {

 print an asterisk

 }

 start next row

}

```
#include <stdio.h>
/* Print an m-by-n rectangle of asterisks */
int main()
{
    int rowc, colc, numrow, numcol;

    printf("\nEnter width: ");
    scanf("%d", &numcol);
    printf("\nEnter height: ");
    scanf("%d", &numrow);

    rowc = 0;
    while (rowc < numrow)
    {
        for (colc=0; colc < numcol; colc++)
        {
            printf("***");
        }
        printf("\n");
        rowc++;
    }
    return 0;
}
```

76

Variation: rect3.c

Print an m by n rectangle of asterisks

input width and height

for each row

{

 for each column in the current row

 {

 print an asterisk

 }

 start next row

}

```
#include <stdio.h>
/* Print an m-by-n rectangle of asterisks */
int main()
{
    int rowc, colc, numrow, numcol;

    printf("\nEnter width: ");
    scanf("%d", &numcol);
    printf("\nEnter height: ");
    scanf("%d", &numrow);

    for (rowc=0; rowc < numrow; rowc++)
    {
        colc = 0;
        while (1)
        {
            printf("***");
            colc++;
            if (colc == numcol)
            { break; }
        }
        printf("\n");
    }
    return 0;
}
```

77

Variation: rect3.c (cont)

Print an m by n rectangle of asterisks

input width and height

for each row

{

 for each column in the current row

 {

 print an asterisk

 }

 start next row

}

The innermost enclosing loop for this break is the while-loop

```
#include <stdio.h>
/* Print an m-by-n rectangle of asterisks */
int main()
{
    int rowc, colc, numrow, numcol;

    printf("\nEnter width: ");
    scanf("%d", &numcol);
    printf("\nEnter height: ");
    scanf("%d", &numrow);

    for (rowc=0; rowc < numrow; rowc++)
    {
        colc = 0;
        while (1)
        {
            printf("***");
            colc++;
            if (colc == numcol)
            { break; }
        }
        printf("\n");
    }
    return 0;
}
```

78

Reading

- King
 - Chapter 6, except Section 6.2
- Deitel and Deitel
 - Chapter 3, Section 3.7
 - Chapter 4,
 - Sections 4.1 to 4.6
 - Sections 4.8 to 4.11

79