



FACULTY OF ENGINEERING AND TECHNOLOGY

COMPUTER SCIENCE DEPARTMENT

COMP1310 Introduction to Computer and Computing Ethics

STUDENTS-HUB.com

REPETITION AND LOOP STATEMENTS

Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022 Uploaded By: anonymous

STUDENTS-HUB.com

Loops

- We use loops to indicate a part of the code that needs to be repeated a certain number of times.
- Types of loops in C:
- 1. while loops
- 2. for loops
- 3. do-while loops

Controlling Loops

- Loops can be controlled in several ways:
- 1. Counter controlled loops: loops that count up or down until a specified value of the counter is reached.
- 2. Event controlled loops: loops that stop when a specific event happens, such as the input reaching 0, or another specific value.
- 3. Result controlled loops: loops that stop when a test determines that the loop reached the desired result, such as a numerical approximation.

initialization
while(condition) {
 statement(s)
 update statement

}

}



The while loop – Counter Controlled

- Let's write a program that calculates the average of students' grades.
- The program will ask the user for the number of students they have in class.
- We can use this number as a counter to know how many loops we need to make.
- When we reach the number that the user entered, we can stop.

The while loop – Counter Controlled

```
#include <stdio.h>
 1
 2
 3
    void main() {
         printf("What is the number of values you wish to enter? ");
 4
 5
         int count;
         scanf("%d", &count);
 6
 7
 8
         int i = 0;
         double sum = 0;
 9
10
         while(i < count) {</pre>
11
             printf("Please enter a grade: ");
12
             double grade;
13
             scanf("%lf", &grade);
14
15
             sum += grade;
16
             i ++;
17
         }
18
         if(count) {
19
             printf("The average of the grades you entered is %.2f.", sum/count);
20
         }
21
22
```

STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

The while Loop – Event Controlled

- Let's write a program that calculates the average of students' grades.
- This time, we will assume that the user does not initially know how many students they have in class.
- We will ask the user to enter the value O when they wish to exit the loop.
- This method assumes that there will be no 0 grades in the entered values.
- If we know that 0 is a possible grade, we need to pick a different condition to stop, such as -1.

The while Loop – Event Controlled

```
#include <stdio.h>
        1
        2
        3
            void main() {
        4
        5
                int \mathbf{i} = \mathbf{0};
        6
                double sum = 0;
        7
                printf("Please enter a grade (enter 0 to stop): ");
        8
                double grade;
        9
                scanf("%lf", &grade);
       10
       11
       12
                while(grade != 0) {
       13
                     sum += grade;
       14
                     printf("Please enter a grade (enter 0 to stop): ");
       15
                     scanf("%lf", &grade);
       16
                     i++;
       17
       18
                if(i != 0) \{
       19
       20
                     printf("The average of the grades you entered is %.2f.", sum/i);
       21
                } else {
       22
                     printf("The average of the grades you entered is 0.",);
       23
       24
DENTS-HUB.com
                        Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022
```

ST

The do-while loops differs form the regular while loop in that we do the statements initially, then we check the while condition to decide if we want to continue or exit the loop.

do {

statements

} while(condition);

The do-while loops differs form the regular while loop in that we do the statements initially, then we check the while condition to decide if we want to continue or exit the loop.

do {

statements **execute these statements first**

} while(condition);

The do-while loops differs form the regular while loop in that we do the statements initially, then we check the while condition to decide if we want to continue or exit the loop.

do {



The do-while loops differs form the regular while loop in that we do the statements initially, then we check the while condition to decide if we want to continue or exit the loop.

do {



■ *do-while* statements are less frequently used than other types of loops.

do-while Example

- Write a program that reads numbers from the user and prints out the sum of these numbers.
- Use the do-while, to make sure at least one number is read from the user.

do-while Example

```
#include <stdio.h>
 1
 2
 3
     void main()
 4
     {
 5
         double number, sum = 0;
 6
 7
         do
 8
         {
             printf("Enter a number: ");
 9
             scanf("%lf", &number);
10
11
             sum += number;
12
13
         while(number != 0.0);
14
15
         printf("Sum = %.2lf", sum);
16
```

DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

Uploaded By: anonymous

S

The for Loop

}

for (initial expression; condition; update statement) {
 statements;

The for Loop

}

for (initial expression; condition; update statement) {
 statements;

- If we omit the initial expression, all initialization must be done before the loop
- If we omit the condition, we need a break inside the loop
- If we omit the update statement, we need to update the values before the end of the loop

Write a loop that prints the numbers from 0 to 9, each on a new line

Write a loop that prints the numbers from 0 to 9, each on a new line

```
#include <stdio.h>
 1
 2
3
     void main() {
 4
         int i = 0;
 5
6
7
         while(i < 10) {
              printf("%d\n", i);
 8
              i++;
 9
         }
10
         printf("Bye!");
11
```

Write a loop that prints the numbers from 0 to 9, each on a new line

```
#include <stdio.h>
 1
 2
3
                                                 #include <stdio.h>
                                             1
     void main() {
                                             2
 4
         int i = 0;
                                             3
                                                 void main() {
 5
6
                                                      for(int i = 0; i < 10; i++) {</pre>
                                             4
         while(i < 10) {
                                             5
6
                                                          printf("%d\n", i);
 7
              printf("%d\n", i);
 8
              i++;
                                             7
                                                      printf("Bye!");
 9
          }
                                             8
         printf("Bye!");
10
11
```

Write a loop that prints the numbers from 0 to 9, each on a new line



Write a loop that prints the numbers from 0 to 9, each on a new line



Write a loop that prints the numbers from 0 to 9, each on a new line



break and continue

- break and continue are reserved words that we need to change the flow of loop statements.
- break and continue statements are usually used with an if statement inside the loop.
- A break statements takes the control out of the loop. This means that if we reach a break statement, the next statement the compiler executes will be the one after the loop's end.
- A continue statement takes the control to the beginning of the loop. This means that if we reach a continue statement, the next statement the compiler executes will be the first statement inside the loop.

break and continue

Behavior of the *break* statement

Behavior of the *continue* statement



Images from https://www.programiz.com/c-programming/c-break-continue-statement

break – Example

Write a loop that prints the numbers from 0 to 5

```
#include <stdio.h>
 1
 2
 3
     void main() {
         int i = 0;
 4
 5
         while(i < 10) {</pre>
 6
              printf("%d\n", i);
 7
 8
              if(i == 5) \{
 9
                  break;
10
11
              i++;
12
13
         printf("Bye!");
14
     }
```

continue – Example

Write a loop that reads 10 numbers from the user and sums the positive numbers only

```
#include <stdio.h>
void main() {
   int number, sum = 0.0;
   for (int i = 0; i < 10; i++) {
      printf("Enter a number: ");
      scanf("%d", &number);
      if (number < 0) {
         continue;
      }
      sum += number;
   }
   printf("Sum = %d", sum);
}
```

Uploaded By: anonymous

DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

S

- When we have a nested loop, the inner loop needs to finish completely before moving on to the next round in the outer loop.
- Nested loops can be of the same type (e.g. two for loops) or from different types (a for loop and a while loop).
- There can be more than two nested loops within each others.
- HOWEVER, be careful that nested loops mean longer processing time.

```
What is the output of the following code:
#include <stdio.h>
 1
 2
 3
    void main() {
 4
 5
         printf("i\tj\n");
         for (int i = 1; i < 4; i++) {</pre>
 6
 7
             for (int j = i; j > 0; j--) {
 8
                 printf("%d\t%d\n", i, j);
 9
             printf("-----\n");
10
11
         }
12
    }
```

```
What is the output of the following code:
#include <stdio.h>
 1
 2
 3
    void main() {
 4
 5
        printf("i\tj\n");
         for (int i = 1; i < 4; i++) {
 6
 7
             for (int j = i; j > 0; j--) {
 8
                 printf("%d\t%d\n", i, j);
 9
             printf("-----\n");
10
11
         }
12
    }
```

Uploaded By: anonymous

ST

action

```
What is the output of the following code:
#include <stdio.h>
 1
 2
 3
    void main() {
 4
 5
        printf("i\tj\n");
         for (int i = 1; i < 4; i++) {
 6
 7
             for (int j = i; j > 0; j--) {
 8
                 printf("%d\t%d\n", i, j);
 9
10
             printf("-----\n");
11
         }
12
    }
```

Uploaded By: anonymous

ST

What is the output of the following code: #include <stdio.h> 1 2 3 void main() { 4 5 printf("i\tj\n"); 6 for (int i = 1; i < 4; i++) {</pre> 7 for (int j = i; j > 0; j--) { 8 printf("%d\t%d\n", i, j); 9 printf("-----\n"); 10 11 } 12 }

DENTS-HUB.com Dima Taji - Birzeit University - COMP1310 - Second Semester 2021/2022 Uploaded By: anonymous

action

1

ST

i j action

```
What is the output of the following code:
#include <stdio.h>
 1
 2
 3
    void main() {
 4
 5
         printf("i\tj\n");
 6
         for (int i = 1; i < 4; i++) {</pre>
 7
             for (int j = i; j > 0; j--) {
 8
                 printf("%d\t%d\n", i, j);
 9
10
             printf("-----\n");
11
         }
12
     }
```
i j action 1 1 print (j)

```
What is the output of the following code:
#include <stdio.h>
 1
 2
 3
    void main() {
 4
 5
         printf("i\tj\n");
 6
         for (int i = 1; i < 4; i++) {</pre>
 7
             for (int j = i; j > 0; j--) {
 8
                 printf("%d\t%d\n", i, j);
 9
10
             printf("-----\n");
11
         }
12
     }
```

What is the output of the following code: #include <stdio.h> 1 2 3 void main() { 4 5 printf("i\tj\n"); 6 for (int i = 1; i < 4; i++) {</pre> 7 for (*int* j = i; j > 0; j--) { 8 printf("%d\t%d\n", i, j); 9 printf("-----\n"); 10 11 } 12 }

action 1 1 print (j) 0

ST DENTS-HUB.com Dima Taji - Birzeit University - COMP1310 - Second Semester 2021/2022



j action 1 print (j) 0 exit (j)

1

```
What is the output of the following code:
#include <stdio.h>
 1
 2
 3
     void main() {
 4
         printf("i\tj\n");
 5
 6
         for (int i = 1; i < 4; i++) {</pre>
 7
             for (int j = i; j > 0; j--) {
 8
                 printf("%d\t%d\n", i, j);
 9
             printf("-----\n");
10
11
         }
12
     }
```



1

STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022





1

2

1

0



```
j action
1 print (j)
0 exit (j)
print (i)
2
```

1

2





		i	j	action
		1	1	print (j
			0	exit (j)
	What is the output of the following code:			nrint (i
1	<pre>#include <stdio.h></stdio.h></pre>			
2		2	2	print (j
3	<pre>void main() {</pre>		1	print (i)
4			Ŧ	print ()
5	<pre>printf("i\tj\n");</pre>			
6	for (int i = 1; i < 4; i++) {			
7	for (<i>int</i> j = i; j > 0; j) {			
8	<pre>printf("%d\t%d\n", i, j);</pre>			
9	}			
10	<i>printf</i> ("\n");			
11	}			
12	}			

		i	j	action
		1	1	print (j)
			0	exit (j)
	What is the output of the following code:			nrint (i)
1	<pre>#include <stdio.h></stdio.h></pre>			
2		2	2	print (j)
3	<pre>void main() {</pre>		1	print (i)
4			–	print ()
5	<pre>printf("i\tj\n");</pre>		0	
6	for (int i = 1; i < 4; i++) {			
7	for (<i>int</i> j = i; j > 0; j) {			
8	<pre>printf("%d\t%d\n", i, j);</pre>			
9	}			
10	<i>printf</i> ("\n");			
11	}			
12	}			

12 }

		i	j	action
		1	1	print (j)
			0	exit (j)
	What is the output of the following code:			nrint (i)
1	<pre>#include <stdio.h></stdio.h></pre>			princ (i)
2		2	2	print (j)
3	<pre>void main() {</pre>		1	print (i)
4			T	princ ()
5	<pre>printf("i\tj\n");</pre>		0	exit (i)
6	for (int $i = 1; i < 4; i++$) {			0/
7	for (<i>int</i> j = i; j > 0; j) {			
8	<pre>printf("%d\t%d\n", i, j);</pre>			
9	}			
10	<i>printf</i> ("\n");			
11	}			

		i	j	action
		1	1	print (j)
			0	exit (j)
	What is the output of the following code:			nrint (i)
1	<pre>#include <stdio.h></stdio.h></pre>			
2		2	2	print (j)
3	<pre>void main() {</pre>		1	$\mathbf{p}_{\mathbf{r}}$
4			T	print ())
5	<pre>printf("i\tj\n");</pre>		0	exit (j)
6	for (int i = 1; i < 4; i++) {			0,
7	for (<i>int</i> j = i; j > 0; j) {			print (i)
8	<pre>printf("%d\t%d\n", i, j);</pre>			
9	}			
10	<i>printf</i> ("\n");			
11	}			
12	}			

ST DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

		I	J	action
		1	1	print (j)
			0	exit (j)
I	What is the output of the following code:			print (i)
1	<pre>#include <stdio.h></stdio.h></pre>			
2		2	2	print (j)
3	<pre>void main() {</pre>		1	print (i)
4			Ŧ	print ()
5	<pre>printf("i\tj\n");</pre>		0	exit (j)
6	for (int i = 1; i < 4; i++) {			
7	for (<i>int</i> j = i; j > 0; j) {			print (I)
8	<pre>printf("%d\t%d\n", i, j);</pre>	З		
9	}	0		
10	<i>printf</i> ("\n");			
11	}			
12	}			

		I	J	action
		1	1	print (j)
			0	exit (j)
	What is the output of the following code:			print (i)
1	<pre>#include <stdio.h></stdio.h></pre>			
2		2	2	print (j)
3	<pre>void main() {</pre>		1	print (i)
4			–	print ()
5	<pre>printf("i\tj\n");</pre>		0	exit (j)
6	for (<i>int</i> i = 1; i < 4; i++) {			
7	for (<i>int</i> j = i; j > 0; j) {			print (I)
8	<i>printf</i> ("%d\t%d\n", i, j);	З	З	
9	}	5	5	
10	<pre>printf("\n");</pre>			
11	}			
12	}			

		I	J	action
		1	1	print (j)
			0	exit (j)
	What is the output of the following code:			print (i)
1	<pre>#include <stdio.h></stdio.h></pre>			
2		2	2	print (j)
3	<pre>void main() {</pre>		1	print (i)
4			Ŧ	print ()
5	<pre>printf("i\tj\n");</pre>		0	exit (j)
6	for (<i>int</i> i = 1; i < 4; i++) {			
7	for (<i>int</i> j = i; j > 0; j) {			print (I)
8	<pre>printf("%d\t%d\n", i, j);</pre>	З	З	nrint (i)
9	}	5	0	princ ()
10	<pre>printf("\n");</pre>			
11	}			
12	}			

		I	J	action
		1	1	print (j)
			0	exit (j)
	What is the output of the following code:			print (i)
1	<pre>#include <stdio.h></stdio.h></pre>		_	pc (.)
2		2	2	print (j)
3	<pre>void main() {</pre>		1	nrint (i)
4			–	princ ()
5	<pre>printf("i\tj\n");</pre>		0	exit (j)
6	for (int i = 1; i < 4; i++) {			
7	for (<i>int</i> j = i; j > 0; j) {			print (I)
8	<i>printf</i> ("%d\t%d\n", i, j);	3	З	nrint (i)
9	}	0	0	princ ()
10	<pre>printf("\n");</pre>		2	
11	}			
12	}			

		I	J	action
		1	1	print (j)
			0	exit (j)
	What is the output of the following code:			print (i)
1	<pre>#include <stdio.h></stdio.h></pre>			
2		2	2	print (j)
3	<pre>void main() {</pre>		1	print (i)
4			–	print ()
5	<pre>printf("i\tj\n");</pre>		0	exit (j)
6	for (int i = 1; i < 4; i++) {			
7	for (<i>int</i> j = i; j > 0; j) {			print (i)
8	<pre>printf("%d\t%d\n", i, j);</pre>	З	З	nrint (i)
9	}	0	0	princ ()
10	<i>printf</i> ("\n");		2	print (j)
11	}			
12	}			

action
print (j)
exit (j)
print (i)
P
print (j)
nrint (i)
princ ()
exit (j)
print (I)
nrint (i)
princ ()
print (j)

		I	J	action
		1	1	print (j)
			0	exit (j)
	What is the output of the following code:			print (i)
1	<pre>#include <stdio.h></stdio.h></pre>	•	-	
2		2	2	print (j)
3	<pre>void main() {</pre>		1	nrint (i)
4			-	print ()
5	<pre>printf("i\tj\n");</pre>		0	exit (j)
6	for (<i>int</i> i = 1; i < 4; i++) {			• • • • • •
7	for (<i>int</i> j = i; j > 0; j) {			print (i)
8	<pre>printf("%d\t%d\n", i, j);</pre>	З	З	nrint (i)
9	}	0	0	princ ()
10	<pre>printf("\n");</pre>		2	print (j)
11	}		1	print (i)
12	}		T	print (J)

	I	J	action
	1	1	print (j)
		0	exit (j)
 What is the output of the following code 	de:		nrint (i)
<pre>1 #include <stdio.h></stdio.h></pre>			
2	2	2	print (j)
<pre>3 void main() {</pre>		1	print (j)
4		-	1 0/
5 printf("1\tj\n");		0	exit (j)
6 for (int $i = 1; i < 4; i++$) {			print (i)
7 for (int $j = i; j > 0; j$	-) {		print (i)
8 <i>printf</i> ("%d\t%d\n", i,	j); 3	3	print (i)
9 }	C	U	
10 <i>printf("\n");</i>		2	print (j)
11 }		1	print (i)
12 }		Ŧ	print ()
		0	

ST DENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

		I	J	action
		1	1	print (j)
			0	exit (j)
	What is the output of the following code:			nrint (i)
1	<pre>#include <stdio.h></stdio.h></pre>			princ (i)
2		2	2	print (j)
3	<pre>void main() {</pre>		1	print (i)
4			_	p()
5	<pre>printf("i\tj\n");</pre>		0	exit (j)
6	for (int i = 1; i < 4; i++) {			print (i)
7	for (int $j = i; j > 0; j)$ {			print (i)
8	<pre>printf("%d\t%d\n", i, j);</pre>	3	3	print (j)
10			0	print (i)
10	<i>printr("(n");</i>		2	print ()
11 12	} }		1	print (j)
and Real	5		0	exit (i)
			v	

	I	J	action
	1	1	print (j)
		0	exit (j)
What is the output of the following code:			print (i)
1 #include <stdio.h> 2</stdio.h>	2	2	print (j)
3 void main() {		1	print (j)
<pre>5 printf("i\tj\n");</pre>		0	exit (j)
6 for $(1nt 1 = 1; 1 < 4; 1++) $ { 7 for $(1nt j = 1; j > 0; j) $ {			print (i)
8	3	3	print (j)
10 <i>printf("\n");</i>		2	print (j)
11 } 12 }		1	print (j)
		0	exit (j)
			print (i)

ST DENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

I	J	action
1	1	print (j)
	0	exit (j)
		print (i)
2	2	print (j)
	1	print (j)
	0	exit (j)
		print (i)
3	3	print (j)
	2	print (j)
	1	print (j)
	0	exit (j)
		print (i)
4		
	1 2 3	I J 1 1 0 2 2 1 0 3 3 2 1 0 4

STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

	İ	Ĵ	action
	1	1	print (j)
		0	exit (j)
What is the output of the following code:			print (i)
2	2	2	print (j)
3		1	print (j)
<pre>5 printf("i\tj\n");</pre>		0	exit (j)
$\begin{array}{cccccccccccccccccccccccccccccccccccc$			print (i)
8	3	3	print (j)
10 <i>printf("\n");</i>		2	print (j)
11 } 12 }		1	print (j)
		0	exit (j)
			print (i)
	4		exit (i)
DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022			Uploaded By: anonymous

ST



Loops – Examples: Perfect Number

- Let's write the code that reads a number from the user and finds if the number is a perfect number.
- A perfect number is the number that the sum of all its divisors, excluding itself, equals the number itself.
- For example:

6 is a perfect number because 1 + 2 + 3 = 6.

28 is a perfect number because 1 + 2 + 4 + 7 + 14 = 28.

30 is not a perfect number because 1 + 2 + 3 + 5 + 6 + 10 + 15 = 42.

Loops – Examples: Perfect Number

```
#include <stdio.h>
                   1
                   2
                      #include <math.h>
                   3
                   4
                      void main() {
                   5
                   6
                           printf("Please enter an integer number: \n");
                   7
                           int num;
                           scanf("%d", &num);
                   8
                   9
                  10
                           int i = 1, sum = 0;
                  11
                           while (i <= num/2) {
                  12
                               if(num % i == 0) {
                  13
                                   sum += i;
                  14
                  15
                  16
                               i++;
                  17
                           }
                  18
                  19
                           if(num == sum) {
                               printf("%d is a perfect number.\n", num);
                  20
                           } else {
                  21
                               printf("%d is not a perfect number.\n", num);
                  22
                           }
                  23
                  24
DENTS-HUB.com
```

Dima Taji - Birzeit University - COMP1310 - Second Semester 2021/2022

Uploaded By: anonymous

ST

Loops – Examples: Narcissistic Numbers

- Let's write the code that reads a number from the user and finds if the number is a narcissistic number.
- A narcissistic number is the number that the sum of each of its digit raised to the nth, where n is the number of digits in the number, equals the number itself.
- For example:

4 is a narcissistic number because $4^1 = 4$.

153 is a narcissistic number because $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$.

8208 is a narcissistic number because $8^4 + 2^4 + 0^4 + 8^8 = 4096 + 16 + 0 + 4096 = 8208$.

25 is not a narcissistic number because $2^2 + 5^2 = 4 + 25 = 29$.

DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

Loops – Examples: Narcissistic Numbers

```
#include <stdio.h>
    #include <math.h>
 2
 3
 4
    void main() {
 5
 6
         printf("Please enter an integer number: \n");
 7
         int num:
 8
         scanf("%d", &num);
 9
         int digits = 0;
10
11
         int temp_num = num;
12
13
         while(temp_num != 0){
             digits += 1;
14
15
             temp_num /= 10;
16
         }
17
```

DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

Uploaded By: anonymous

S

Loops – Examples: Narcissistic Numbers

```
18
         int sum = 0;
19
20
         temp num = num;
21
22
         while(temp_num != 0){
23
             int digit = temp_num % 10;
24
             sum += pow(digit, digits);
25
             temp_num /= 10;
26
         }
27
28
         if(num == sum) {
             printf("%d is a narcissistic number.\n", num);
29
30
         } else {
31
             printf("%d is not a narcissistic number.\n", num);
32
33
     }
```

STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

- Let's write the code prints out the first 20 narcissistic numbers.
- We can create a function that takes a number and returns 1 if it is a narcissistic number, and 0 if it is not.
- We can create a result-controlled loop that exits if we reached 20 narcissistic numbers.

```
#include <stdio.h>
 1
    #include <math.h>
 2
 3
    int is_narcissistic(int);
 4
 5
 6
    void main() {
 7
 8
        int counter = 0;
 9
         int num = 0;
10
        while(counter != 20) {
11
             if(is_narcissistic(num)) {
12
                 printf("%d is a narcissistic number.\n", num);
13
14
                 counter += 1;
15
16
             num += 1;
17
18
    }
19
```

STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

20	<pre>int is_narcissistic(int num)</pre>	{
21	<pre>int digits = 0;</pre>	
22	<pre>int temp_num = num;</pre>	
23		
24	<pre>while(temp_num != 0){</pre>	
25	digits += 1;	
26	temp_num $/= 10;$	
27	}	
28		
29	<pre>int sum = 0;</pre>	
30		
31	<pre>temp_num = num;</pre>	
32		

DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

S



STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

- Can we rewrite the main part of the previous example using a for loop instead of a while loop?
- To do that, we need to determine the following:
 - What are the initializing statements?
 - What is the loop condition?
 - What are the update statements?

```
#include <stdio.h>
 1
     #include <math.h>
 2
 3
     int is narcissistic(int);
 4
 5
 6
     void main() {
 7
 8
         for(int counter = 0, num = 0; counter <20; num++) {</pre>
 9
             if(is_narcissistic(num)) {
                  printf("%d is a narcissistic number.\n", num);
10
11
                  counter += 1;
12
13
         }
14
     }
```
```
#include <stdio.h>
 1
     #include <math.h>
 2
 3
     int is narcissistic(int);
 4
 5
 6
     void main() {
 7
 8
         for(int counter = 0, num = 0; counter <20; num++) {</pre>
 9
             if(is narcissistic(num)) {
                  printf("%d is a narcissistic number.\n", num);
10
11
                  counter += 1;
12
13
14
     }
```

This is not an update statement because it doesn't happen in every loop

■ Can we rewrite *for* loop without a condition?

We need to make sure that the compiler knows when to leave the loop.

- We use the break statement.

```
#include <stdio.h>
 1
    #include <math.h>
 2
 3
    int is_narcissistic(int);
 4
 5
 6
    void main() {
 7
         for(int counter = 0, num = 0; ; num++) {
8
             if(is_narcissistic(num)) {
9
10
                 printf("%d is a narcissistic number.\n", num);
11
                 counter += 1;
12
                 if (counter == 20) {
13
                     break;
14
15
16
17
```

```
#include <stdio.h>
 1
 2
    #include <math.h>
 3
    int is_narcissistic(int);
 4
 5
 6
    void main() {
8
         for(int counter = 0, num = 0; ; num++) {
 9
             if(is_narcissistic(num)) {
                 printf("%d is a narcissistic number.\n", num);
10
11
                 counter += 1;
12
                 if (counter == 20) {
13
                     break;
14
15
16
17
```

Notice the semicolons that we need to keep

INCREMENTS

Pre- and post-increments and pre- and post-decrements

STUDENTS-HUB.com

Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022 Uploaded By: anonymous

Pre- and Post-Increment

We have been using expressions that look like

i++ and ++i

to update values in loops.

What do these mean exactly? And how are they different?

++i is called a pre-increment.

■ i++ is called a post-increment.

after

before

Examples

Pre-increment

$$a = ++x * b;$$
 $x = x + 1;$
 $a = x * b;$

Post-increment

$$a = x + * b;$$

 $a = x * b;$
 $x = x + 1;$

DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

Uploaded By: anonymous

S1

Pre- and Post-Decrements

A pre-decrement looks like this:

--x;

and a post-decrement looks like this:

x--;

Everything we will learn about increments works just the same for decrements, with the exception that decrements decrease one from the variable.

Examples

Pre-decrement

$$a = --x * b;$$
 $x = x - 1;$
 $a = x * b;$

Post-decrement

$$a = x - * b;$$

 $a = x * b;$
 $x = x - 1;$

DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

Uploaded By: anonymous

S1

int a=2, b=3, c; c = ++a * b++; printf(``%d\n", a); printf(``%d\n", b); printf(``%d\n", c);



int a=2, b=3, c; c = ++a * b++; printf(``%d\n", a); printf(``%d\n", b); printf(``%d\n", c);

а	b	С
2	3	

а	b	С
2	3	
3	3	

а	b	С
2	3	
3	3	
3	3	9

а	b	С
2	3	
3	3	
3	3	9
3	4	9

int a=2, b=3, c;
c = ++a * b++;
<pre>printf("%d\n", a);</pre>
printf("%d\n", b);
printf("%d\n", c);

а	b	С
2	3	
3	3	
3	3	9
3	4	9

Output:
3
4
9

DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022 Uploaded By: anonymous

S1

int a=4, b=20; b /= ++a; printf(``%d\n", a); printf(``%d\n", b);

а	b

int a=4, b=20; b /= ++a; printf(``%d\n", a); printf(``%d\n", b);

а	b
4	20

int a=4, b=20; b /= ++a printf(``%d\n", a); printf(``%d\n", b);

а	b
4	20
5	20



а	b
4	20
5	20
5	4

STUDENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

Output: 5

4

DENTS-HUB.com Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022

Uploaded By: anonymous

b

20

20

4

S1

int a=4, b=20, c=3; b /= ++a + c--; printf(``%d\n", a); printf(``%d\n", b); printf(``%d\n", c);

а	b	С

int a=4, b=20, c=3; b /= ++a + c--; printf(``%d\n", a); printf(``%d\n", b); printf(``%d\n", c);

а	b	С
4	20	3

а	b	С
4	20	3
5	20	3

ST DENTS-HUB.COM Dima Taji – Birzeit University – COMP1310 – Second Semester 2021/2022



а	b	С
4	20	3
5	20	3



а	b	С
4	20	3
5	20	3
5	2	3

int a=4, b=20, c=3; b /= ++a + c--, printf(``%d\n", a); printf(``%d\n", b); printf(``%d\n", c);

а	b	С
4	20	3
5	20	3
5	2	3
5	2	2

int a=4,	b=20,	c=3;
b /= ++a	+ c;	•
printf(" ⁹	d∖n″,	a);
printf(" ⁹	d∖n″,	b);
printf("%	∦d\n″,	c);

а	d	С
4	20	3
5	20	3
5	2	3
5	2	2

Output:
5
2
2

Final Note on Increments

If the statements consists solely of a pre-increment or decrement, or a post-increment or decrement, their behaviour will be the same.

This means that

is equivalent to

and

is equivalent to

++i;

i++;

i--;

--i;