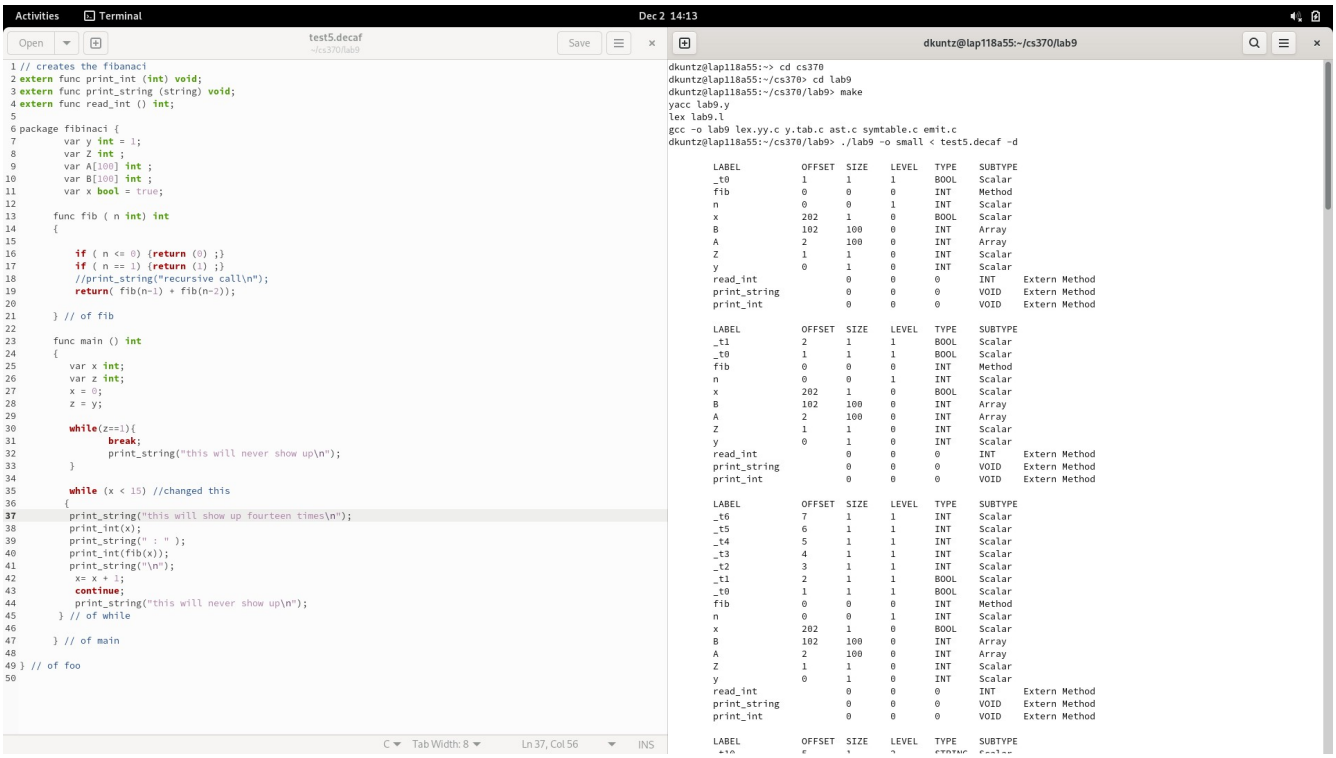
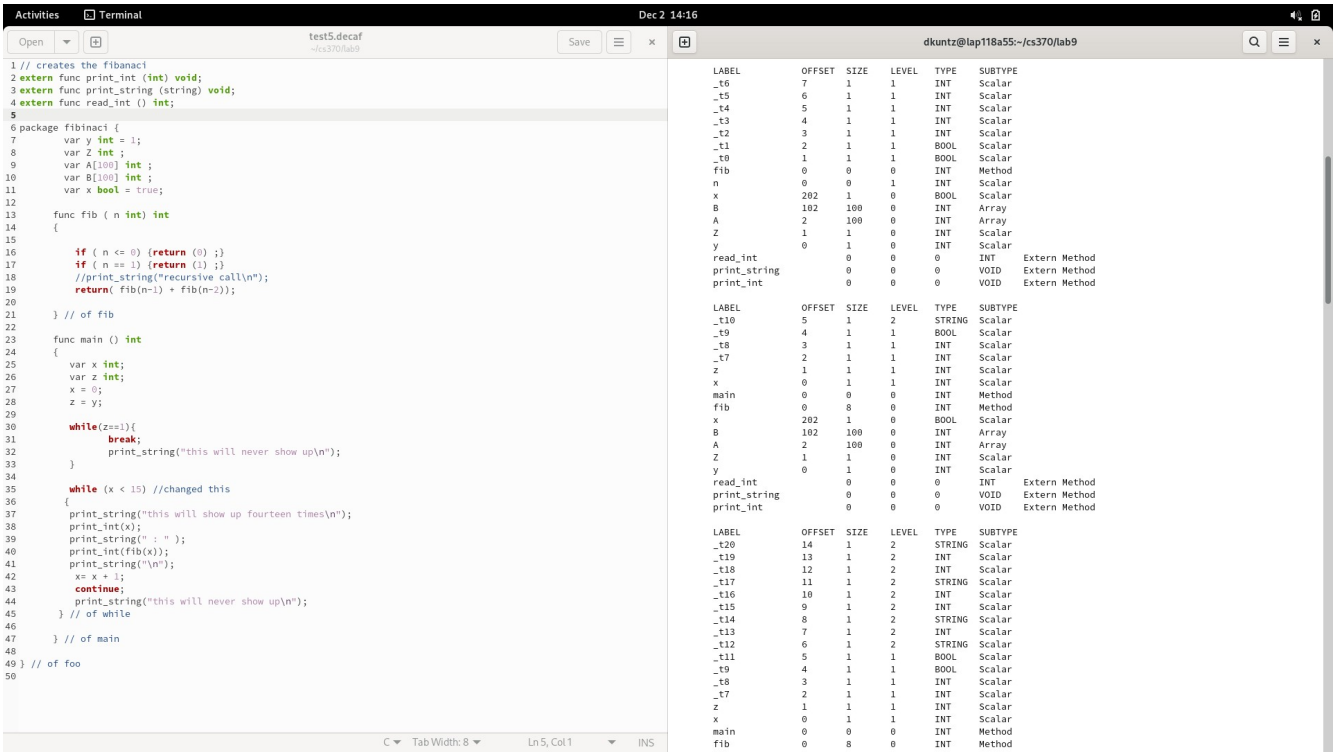
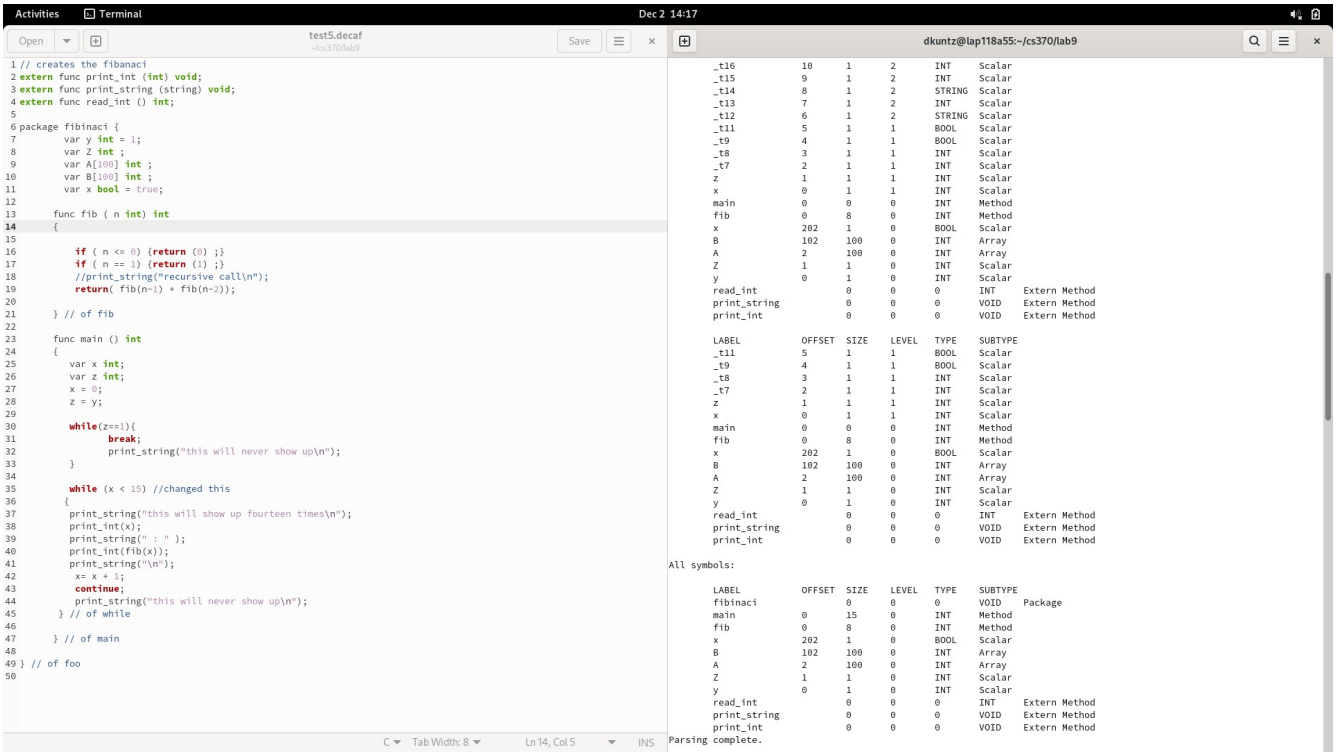
David Kuntz
Fall 2022
CS 370
Lab 9

# Output

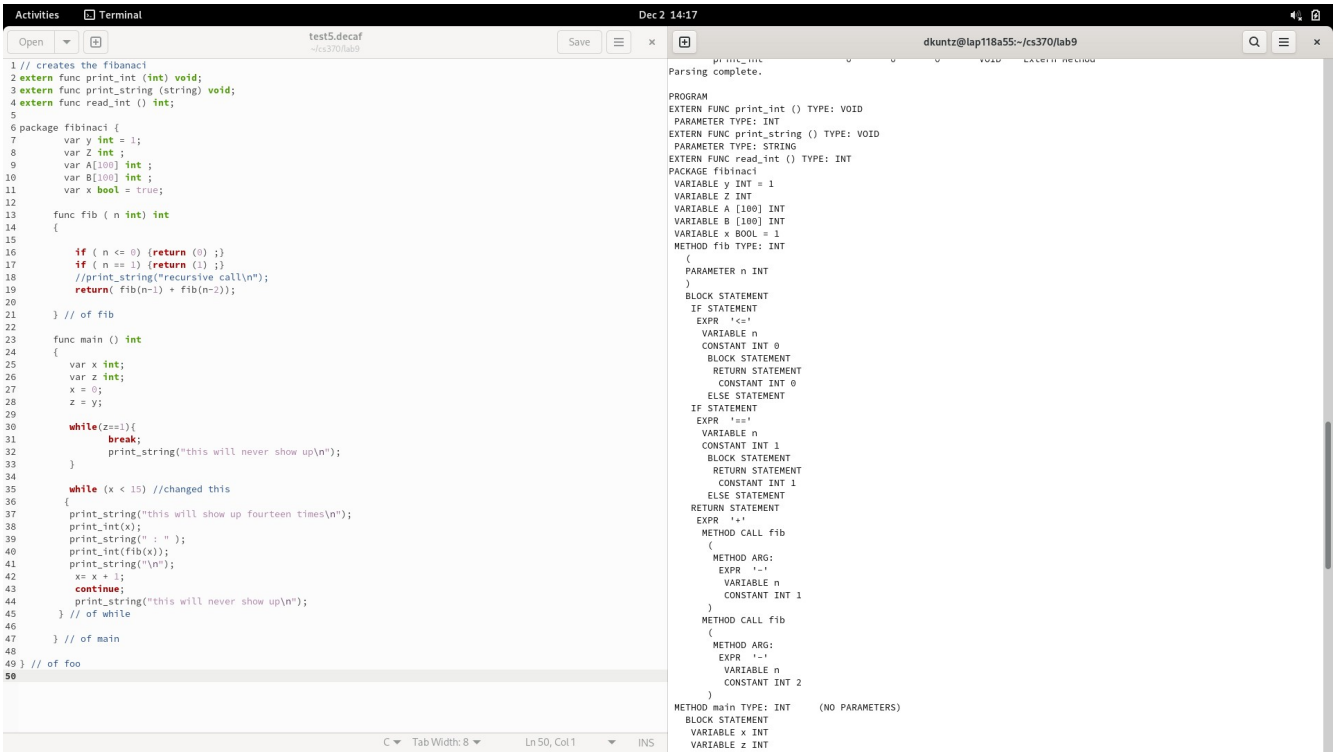



Activities
Terminal
Dec 2 14:16
Open
test5.decaf
~/cs370/lab9
Save
dkuntz@lap118a55:~/cs370/lab9
// creates the fibanaci
extern func print_int (int) void;
extern func print_string (string) void;
extern func read_int () int;
package fibinaci {
var y int = 1;
var Z int ;
var A[100] int ;
var B[100] int ;
var x bool = true;
func fib ( n int) int
{
if ( n <= 0) {return (0) ;}
if ( n == 1) {return (1) ;}
//print_string("recursive call\n");
return( fib(n-1) + fib(n-2));
} // of fib
func main () int
{
var x int;
var z int;
x = 0;
z = y;
while(z==1){
break;
print_string("this will never show up\n");
}
while (x < 15) //changed this
{
print_string("this will show up fourteen times\n");
print_int(x);
print_string(" : " );
print_int(fib(x));
print_string("\n");
x= x + 1;
continue;
print_string("this will never show up\n");
} // of while
} // of main
} // of foo
C
Tab Width: 8
Ln 5, Col 1
INS
LABEL OFFSET SIZE LEVEL TYPE SUBTYPE
_t6 7 1 1 INT Scalar
_t5 6 1 1 INT Scalar
_t4 5 1 1 INT Scalar
_t3 4 1 1 INT Scalar
_t2 3 1 1 INT Scalar
_t1 2 1 1 BOOL Scalar
_t0 1 1 1 BOOL Scalar
fib 0 0 0 INT Method
n 0 0 1 INT Scalar
x 202 1 0 BOOL Scalar
B 102 100 0 INT Array
A 2 100 0 INT Array
Z 1 1 0 INT Scalar
y 0 1 0 INT Scalar
read_int 0 0 0 INT Extern Method
print_string 0 0 0 VOID Extern Method
print_int 0 0 0 VOID Extern Method
LABEL OFFSET SIZE LEVEL TYPE SUBTYPE
_t10 5 1 2 STRING Scalar
_t9 4 1 1 BOOL Scalar
_t8 3 1 1 INT Scalar
_t7 2 1 1 INT Scalar
z 1 1 1 INT Scalar
x 0 1 1 INT Scalar
main 0 0 0 INT Method
fib 0 8 0 INT Method
x 202 1 0 BOOL Scalar
B 102 100 0 INT Array
A 2 100 0 INT Array
Z 1 1 0 INT Scalar
y 0 1 0 INT Scalar
read_int 0 0 0 INT Extern Method
print_string 0 0 0 VOID Extern Method
print_int 0 0 0 VOID Extern Method
LABEL OFFSET SIZE LEVEL TYPE SUBTYPE
_t20 14 1 2 STRING Scalar
_t19 13 1 2 INT Scalar
_t18 12 1 2 INT Scalar
_t17 11 1 2 STRING Scalar
_t16 10 1 2 INT Scalar
_t15 9 1 2 INT Scalar
_t14 8 1 2 STRING Scalar
_t13 7 1 2 INT Scalar
_t12 6 1 2 STRING Scalar
_t11 5 1 1 BOOL Scalar
_t9 4 1 1 BOOL Scalar
_t8 3 1 1 INT Scalar
_t7 2 1 1 INT Scalar
z 1 1 1 INT Scalar
x 0 1 1 INT Scalar
main 0 0 0 INT Method
fib 0 8 0 INT Method

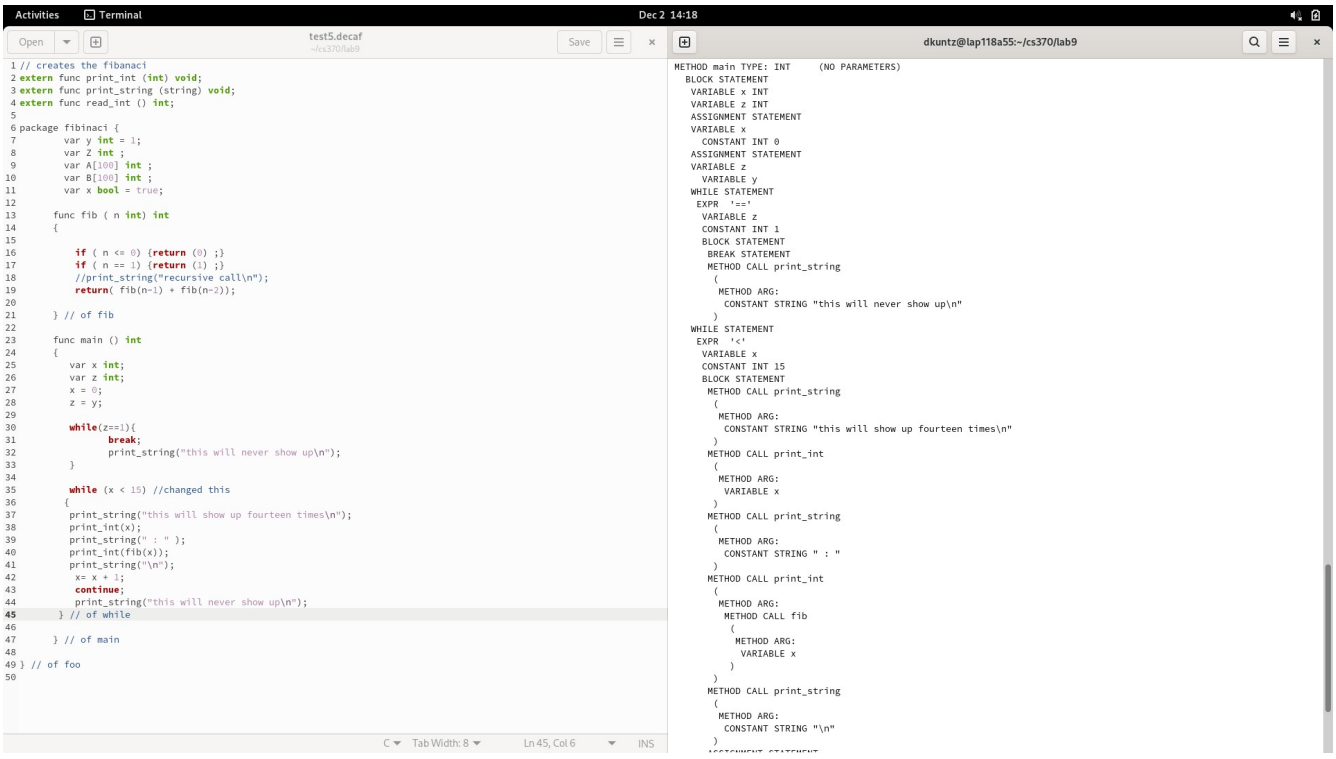
2


Activities
Terminal
Dec 2 14:17
Open
test5.decaf
~/cs370/lab9
Save
dkuntz@lap118a55:~/cs370/lab9
// creates the fibanaci
extern func print_int (int) void;
extern func print_string (string) void;
extern func read_int () int;
package fibinaci {
var y int = 1;
var Z int ;
var A[100] int ;
var B[100] int ;
var x bool = true;
func fib ( n int) int
{
if ( n <= 0) {return (0) ;}
if ( n == 1) {return (1) ;}
//print_string("recursive call\n");
return( fib(n-1) + fib(n-2));
} // of fib
func main () int
{
var x int;
var z int;
x = 0;
z = y;
while(z==1){
break;
print_string("this will never show up\n");
}
while (x < 15) //changed this
{
print_string("this will show up fourteen times\n");
print_int(x);
print_string(" : " );
print_int(fib(x));
print_string("\n");
x= x + 1;
continue;
print_string("this will never show up\n");
} // of while
} // of main
} // of foo
C
Tab Width: 8
Ln 14, Col 5
INS
_t16 10 1 2 INT Scalar
_t15 9 1 2 INT Scalar
_t14 8 1 2 STRING Scalar
_t13 7 1 2 INT Scalar
_t12 6 1 2 STRING Scalar
_t11 5 1 1 BOOL Scalar
_t9 4 1 1 BOOL Scalar
_t8 3 1 1 INT Scalar
_t7 2 1 1 INT Scalar
z 1 1 1 INT Scalar
x 0 1 1 INT Scalar
main 0 0 0 INT Method
fib 0 8 0 INT Method
x 202 1 0 BOOL Scalar
B 102 100 0 INT Array
A 2 100 0 INT Array
Z 1 1 0 INT Scalar
y 0 1 0 INT Scalar
read_int 0 0 0 INT Extern Method
print_string 0 0 0 VOID Extern Method
print_int 0 0 0 VOID Extern Method
LABEL OFFSET SIZE LEVEL TYPE SUBTYPE
_t11 5 1 1 BOOL Scalar
_t9 4 1 1 BOOL Scalar
_t8 3 1 1 INT Scalar
_t7 2 1 1 INT Scalar
z 1 1 1 INT Scalar
x 0 1 1 INT Scalar
main 0 0 0 INT Method
fib 0 8 0 INT Method
x 202 1 0 BOOL Scalar
B 102 100 0 INT Array
A 2 100 0 INT Array
Z 1 1 0 INT Scalar
y 0 1 0 INT Scalar
read_int 0 0 0 INT Extern Method
print_string 0 0 0 VOID Extern Method
print_int 0 0 0 VOID Extern Method
All symbols:
LABEL OFFSET SIZE LEVEL TYPE SUBTYPE
fibinaci 0 0 0 VOID Package
main 0 15 0 INT Method
fib 0 8 0 INT Method
x 202 1 0 BOOL Scalar
B 102 100 0 INT Array
A 2 100 0 INT Array
Z 1 1 0 INT Scalar
y 0 1 0 INT Scalar
read_int 0 0 0 INT Extern Method
print_string 0 0 0 VOID Extern Method
print_int 0 0 0 VOID Extern Method
Parsing complete.

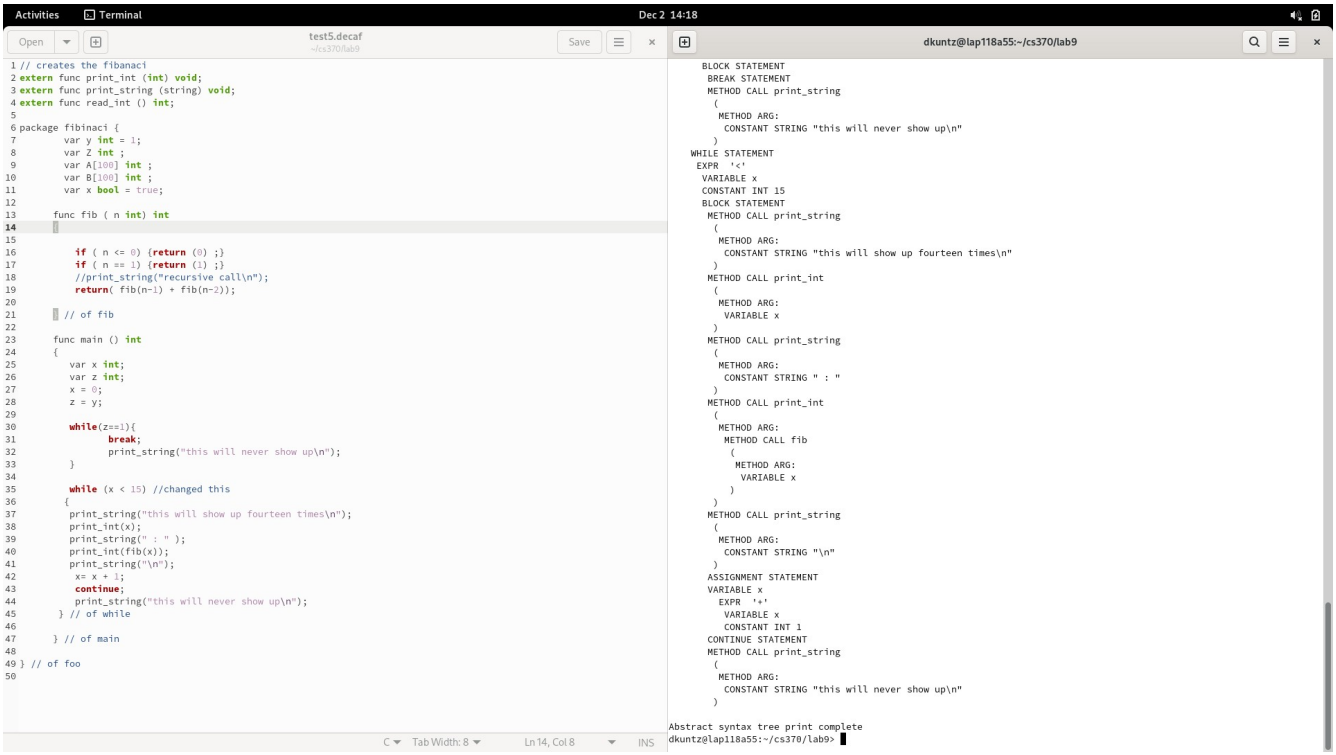
3


Activities
Terminal
Dec 2 14:17
Open
test5.decaf
~/cs370/lab9
Save
dkuntz@lap118a55:~/cs370/lab9
// creates the fibanaci
extern func print_int (int) void;
extern func print_string (string) void;
extern func read_int () int;
package fibinaci {
var y int = 1;
var Z int ;
var A[100] int ;
var B[100] int ;
var x bool = true;
func fib ( n int) int
{
if ( n <= 0) {return (0) ;}
if ( n == 1) {return (1) ;}
//print_string("recursive call\n");
return( fib(n-1) + fib(n-2));
} // of fib
func main () int
{
var x int;
var z int;
x = 0;
z = y;
while(z==1){
break;
print_string("this will never show up\n");
}
while (x < 15) //changed this
{
print_string("this will show up fourteen times\n");
print_int(x);
print_string(" : " );
print_int(fib(x));
print_string("\n");
x= x + 1;
continue;
print_string("this will never show up\n");
} // of while
} // of main
} // of foo
C
Tab Width: 8
Ln 50, Col 1
INS
Parsing complete.
PROGRAM
EXTERN FUNC print_int () TYPE: VOID
PARAMETER TYPE: INT
EXTERN FUNC print_string () TYPE: VOID
PARAMETER TYPE: STRING
EXTERN FUNC read_int () TYPE: INT
PACKAGE fibinaci
VARIABLE y INT = 1
VARIABLE Z INT
VARIABLE A [100] INT
VARIABLE B [100] INT
VARIABLE x BOOL = 1
METHOD fib TYPE: INT
(
PARAMETER n INT
)
BLOCK STATEMENT
IF STATEMENT
EXPR '<='
VARIABLE n
CONSTANT INT 0
BLOCK STATEMENT
RETURN STATEMENT
CONSTANT INT 0
ELSE STATEMENT
IF STATEMENT
EXPR '=='
VARIABLE n
CONSTANT INT 1
BLOCK STATEMENT
RETURN STATEMENT
CONSTANT INT 1
ELSE STATEMENT
RETURN STATEMENT
EXPR '+'
METHOD CALL fib
(
METHOD ARG:
EXPR '-'
VARIABLE n
CONSTANT INT 1
)
METHOD CALL fib
(
METHOD ARG:
EXPR '-'
VARIABLE n
CONSTANT INT 2
)
METHOD main TYPE: INT (NO PARAMETERS)
BLOCK STATEMENT
VARIABLE x INT
VARIABLE z INT

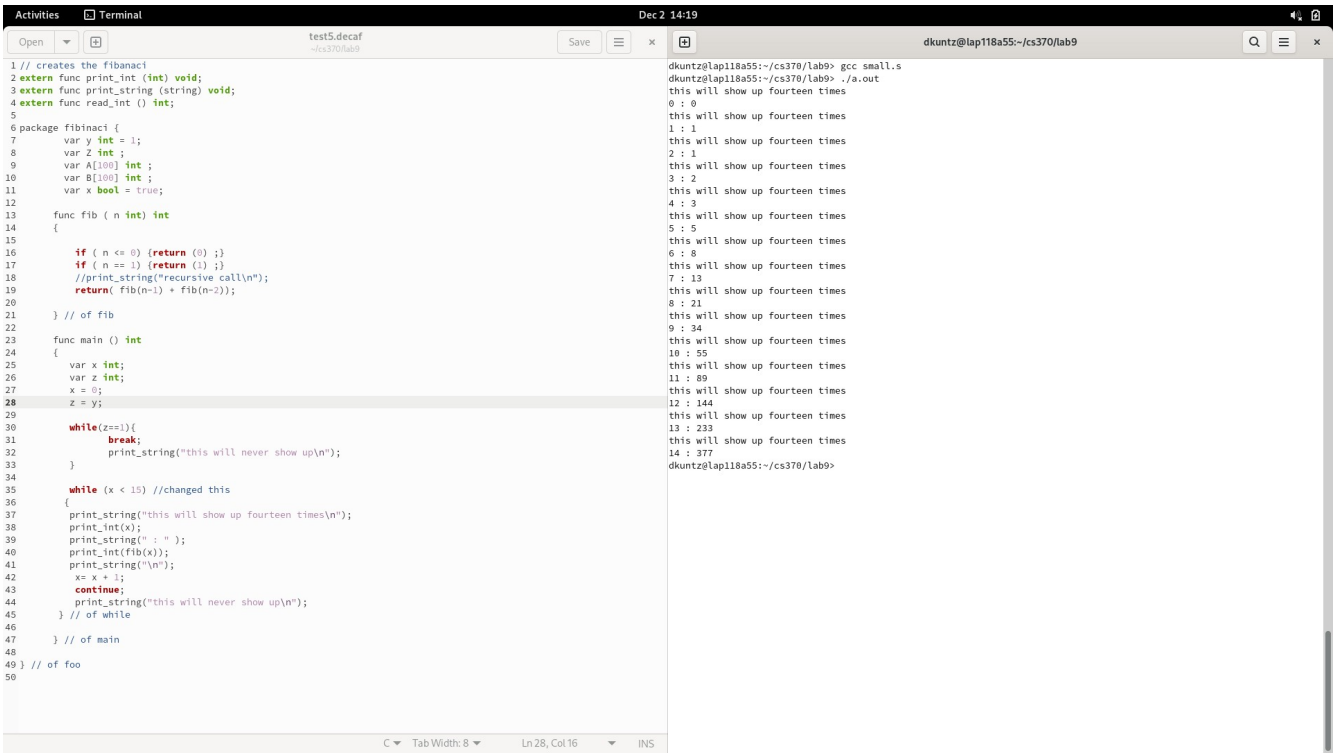
4


Activities
Terminal
Dec 2 14:18
Open
test5.decaf
~/cs370/lab9
Save
dkuntz@lap118a55:~/cs370/lab9
// creates the fibanaci
extern func print_int (int) void;
extern func print_string (string) void;
extern func read_int () int;
package fibinaci {
var y int = 1;
var Z int ;
var A[100] int ;
var B[100] int ;
var x bool = true;
func fib ( n int) int
{
if ( n <= 0) {return (0) ;}
if ( n == 1) {return (1) ;}
//print_string("recursive call\n");
return( fib(n-1) + fib(n-2));
} // of fib
func main () int
{
var x int;
var z int;
x = 0;
z = y;
while(z==1){
break;
print_string("this will never show up\n");
}
while (x < 15) //changed this
{
print_string("this will show up fourteen times\n");
print_int(x);
print_string(" : " );
print_int(fib(x));
print_string("\n");
x= x + 1;
continue;
print_string("this will never show up\n");
} // of while
} // of main
} // of foo
C
Tab Width: 8
Ln 45, Col 6
INS
METHOD main TYPE: INT (NO PARAMETERS)
BLOCK STATEMENT
VARIABLE x INT
VARIABLE z INT
ASSIGNMENT STATEMENT
VARIABLE x
CONSTANT INT 0
ASSIGNMENT STATEMENT
VARIABLE z
VARIABLE y
WHILE STATEMENT
EXPR '=='
VARIABLE z
CONSTANT INT 1
BLOCK STATEMENT
BREAK STATEMENT
METHOD CALL print_string
(
METHOD ARG:
CONSTANT STRING "this will never show up\n"
)
WHILE STATEMENT
EXPR '<'
VARIABLE x
CONSTANT INT 15
BLOCK STATEMENT
METHOD CALL print_string
(
METHOD ARG:
CONSTANT STRING "this will show up fourteen times\n"
)
METHOD CALL print_int
(
METHOD ARG:
VARIABLE x
)
METHOD CALL print_string
(
METHOD ARG:
CONSTANT STRING " : "
)
METHOD CALL print_int
(
METHOD ARG:
METHOD CALL fib
(
METHOD ARG:
VARIABLE x
)
)
METHOD CALL print_string
(
METHOD ARG:
CONSTANT STRING "\n"
)

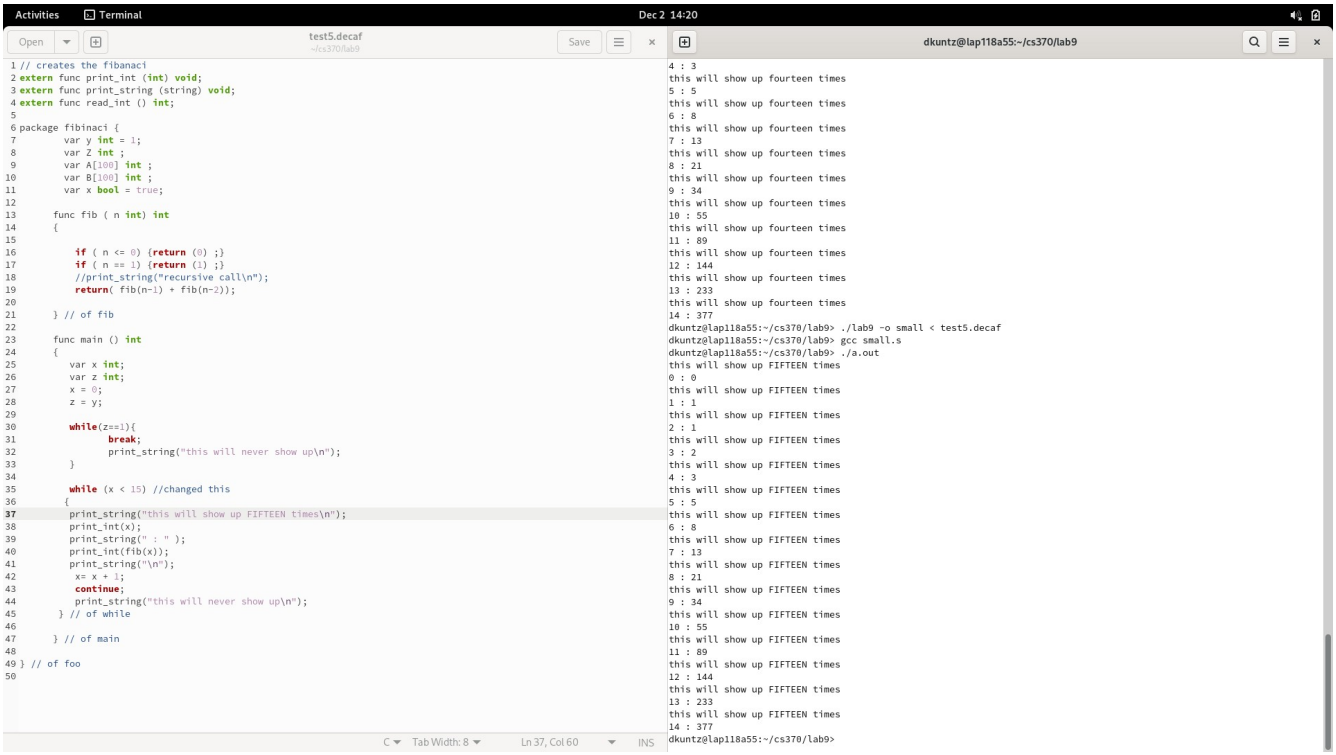
5

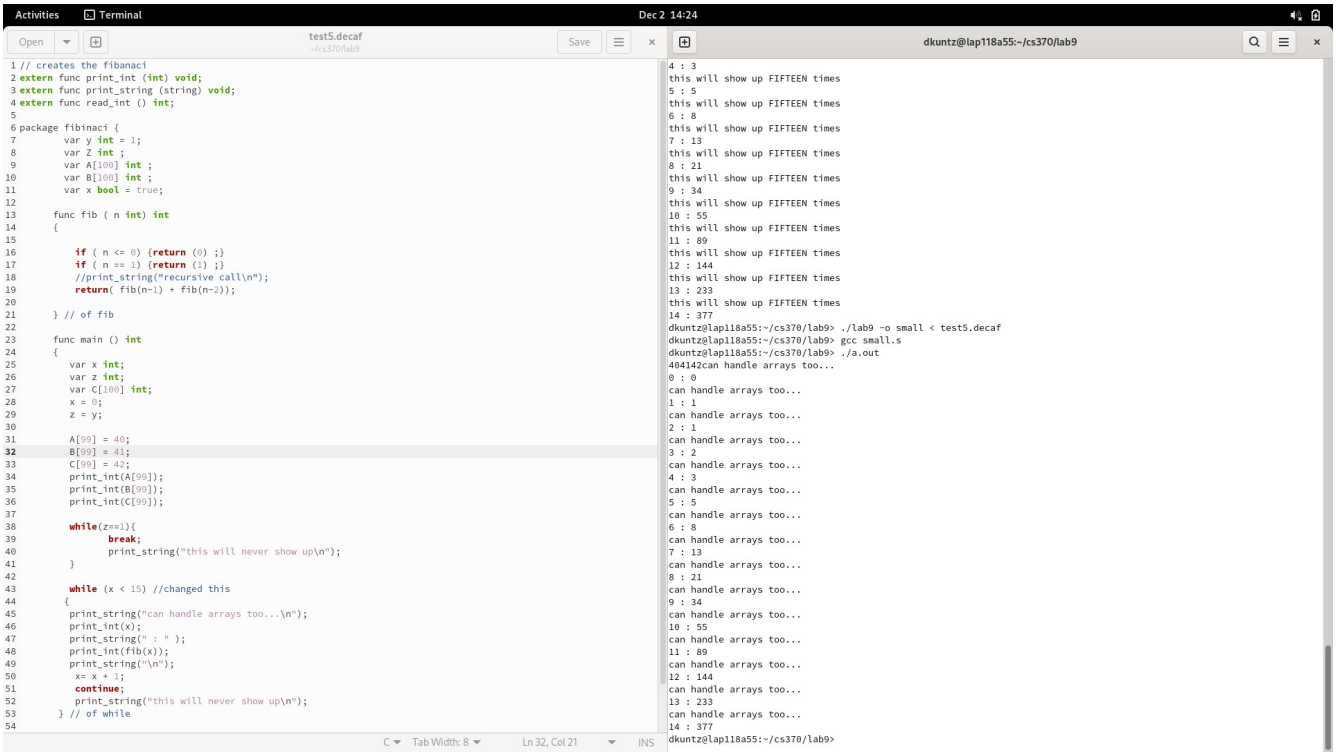

Activities
Terminal
Dec 2 14:18
Open
test5.decaf
~/cs370/lab9
Save
dkuntz@lap118a55:~/cs370/lab9
1 // creates the fibanaci
2 extern func print_int (int) void;
3 extern func print_string (string) void;
4 extern func read_int () int;
5
6 package fibinaci {
7 var y int = 1;
8 var Z int ;
9 var A[100] int ;
10 var B[100] int ;
11 var x bool = true;
12
13 func fib ( n int) int
14
15
16 if ( n <= 0) {return (0) ;}
17 if ( n == 1) {return (1) ;}
18 //print_string("recursive call\n");
19 return( fib(n-1) + fib(n-2));
20
21 } // of fib
22
23 func main () int
24 {
25 var x int;
26 var z int;
27 x = 0;
28 z = y;
29
30 while(z==1){
31 break;
32 print_string("this will never show up\n");
33 }
34
35 while (x < 15) //changed this
36 {
37 print_string("this will show up fourteen times\n");
38 print_int(x);
39 print_string(" : " );
40 print_int(fib(x));
41 print_string("\n");
42 x= x + 1;
43 continue;
44 print_string("this will never show up\n");
45 } // of while
46
47 } // of main
48
49 } // of foo
50
BLOCK STATEMENT
BREAK STATEMENT
METHOD CALL print_string
(
METHOD ARG:
CONSTANT STRING "this will never show up\n"
)
WHILE STATEMENT
EXPR '<'
VARIABLE x
CONSTANT INT 15
BLOCK STATEMENT
METHOD CALL print_string
(
METHOD ARG:
CONSTANT STRING "this will show up fourteen times\n"
)
METHOD CALL print_int
(
METHOD ARG:
VARIABLE x
)
METHOD CALL print_string
(
METHOD ARG:
CONSTANT STRING " : "
)
METHOD CALL print_int
(
METHOD ARG:
METHOD CALL fib
(
METHOD ARG:
VARIABLE x
)
)
METHOD CALL print_string
(
METHOD ARG:
CONSTANT STRING "\n"
)
ASSIGNMENT STATEMENT
VARIABLE x
EXPR '+'
VARIABLE x
CONSTANT INT 1
CONTINUE STATEMENT
METHOD CALL print_string
(
METHOD ARG:
CONSTANT STRING "this will never show up\n"
)
Abstract syntax tree print complete
dkuntz@lap118a55:~/cs370/lab9>
C
Tab Width: 8
Ln 14, Col 8
INS

6


Activities
Terminal
Dec 2 14:19
Open
test5.decaf
~/cs370/lab9
Save
dkuntz@lap118a55:~/cs370/lab9
1 // creates the fibanaci
2 extern func print_int (int) void;
3 extern func print_string (string) void;
4 extern func read_int () int;
5
6 package fibinaci {
7 var y int = 1;
8 var Z int ;
9 var A[100] int ;
10 var B[100] int ;
11 var x bool = true;
12
13 func fib ( n int) int
14 {
15
16 if ( n <= 0) {return (0) ;}
17 if ( n == 1) {return (1) ;}
18 //print_string("recursive call\n");
19 return( fib(n-1) + fib(n-2));
20
21 } // of fib
22
23 func main () int
24 {
25 var x int;
26 var z int;
27 x = 0;
28 z = y;
29
30 while(z==1){
31 break;
32 print_string("this will never show up\n");
33 }
34
35 while (x < 15) //changed this
36 {
37 print_string("this will show up fourteen times\n");
38 print_int(x);
39 print_string(" : " );
40 print_int(fib(x));
41 print_string("\n");
42 x= x + 1;
43 continue;
44 print_string("this will never show up\n");
45 } // of while
46
47 } // of main
48
49 } // of foo
50
dkuntz@lap118a55:~/cs370/lab9> gcc small.s
dkuntz@lap118a55:~/cs370/lab9> ./a.out
this will show up fourteen times
0 : 0
this will show up fourteen times
1 : 1
this will show up fourteen times
2 : 1
this will show up fourteen times
3 : 2
this will show up fourteen times
4 : 3
this will show up fourteen times
5 : 5
this will show up fourteen times
6 : 8
this will show up fourteen times
7 : 13
this will show up fourteen times
8 : 21
this will show up fourteen times
9 : 34
this will show up fourteen times
10 : 55
this will show up fourteen times
11 : 89
this will show up fourteen times
12 : 144
this will show up fourteen times
13 : 233
this will show up fourteen times
14 : 377
dkuntz@lap118a55:~/cs370/lab9>
C
Tab Width: 8
Ln 28, Col 16
INS

7


Activities
Terminal
Dec 2 14:20
Open
test5.decaf
~/cs370/lab9
Save
dkuntz@lap118a55:~/cs370/lab9
// creates the fibanaci
extern func print_int (int) void;
extern func print_string (string) void;
extern func read_int () int;
package fibinaci {
var y int = 1;
var Z int ;
var A[100] int ;
var B[100] int ;
var x bool = true;
func fib ( n int) int
{
if ( n <= 0) {return (0) ;}
if ( n == 1) {return (1) ;}
//print_string("recursive call\n");
return( fib(n-1) + fib(n-2));
} // of fib
func main () int
{
var x int;
var z int;
x = 0;
z = y;
while(z==1){
break;
print_string("this will never show up\n");
}
while (x < 15) //changed this
{
print_string("this will show up FIFTEEN times\n");
print_int(x);
print_string(" : " );
print_int(fib(x));
print_string("\n");
x= x + 1;
continue;
print_string("this will never show up\n");
} // of while
} // of main
} // of foo
C
Tab Width: 8
Ln 37, Col 60
INS
4 : 3
this will show up fourteen times
5 : 5
this will show up fourteen times
6 : 8
this will show up fourteen times
7 : 13
this will show up fourteen times
8 : 21
this will show up fourteen times
9 : 34
this will show up fourteen times
10 : 55
this will show up fourteen times
11 : 89
this will show up fourteen times
12 : 144
this will show up fourteen times
13 : 233
this will show up fourteen times
14 : 377
dkuntz@lap118a55:~/cs370/lab9> ./lab9 -o small < test5.decaf
dkuntz@lap118a55:~/cs370/lab9> gcc small.s
dkuntz@lap118a55:~/cs370/lab9> ./a.out
this will show up FIFTEEN times
0 : 0
this will show up FIFTEEN times
1 : 1
this will show up FIFTEEN times
2 : 1
this will show up FIFTEEN times
3 : 2
this will show up FIFTEEN times
4 : 3
this will show up FIFTEEN times
5 : 5
this will show up FIFTEEN times
6 : 8
this will show up FIFTEEN times
7 : 13
this will show up FIFTEEN times
8 : 21
this will show up FIFTEEN times
9 : 34
this will show up FIFTEEN times
10 : 55
this will show up FIFTEEN times
11 : 89
this will show up FIFTEEN times
12 : 144
this will show up FIFTEEN times
13 : 233
this will show up FIFTEEN times
14 : 377
dkuntz@lap118a55:~/cs370/lab9>


Activities
Terminal
Dec 2 14:24
Open
test5.decaf
~/cs370/lab9
Save
dkuntz@lap118a55:~/cs370/lab9
// creates the fibanaci
extern func print_int (int) void;
extern func print_string (string) void;
extern func read_int () int;
package fibinaci {
var y int = 1;
var Z int ;
var A[100] int ;
var B[100] int ;
var x bool = true;
func fib ( n int) int
{
if ( n <= 0) {return (0) ;}
if ( n == 1) {return (1) ;}
//print_string("recursive call\n");
return( fib(n-1) + fib(n-2));
} // of fib
func main () int
{
var x int;
var z int;
var C[100] int;
x = 0;
z = y;
A[99] = 40;
B[99] = 41;
C[99] = 42;
print_int(A[99]);
print_int(B[99]);
print_int(C[99]);
while(z==1){
break;
print_string("this will never show up\n");
}
while (x < 15) //changed this
{
print_string("can handle arrays too...\n");
print_int(x);
print_string(" : " );
print_int(fib(x));
print_string("\n");
x= x + 1;
continue;
print_string("this will never show up\n");
} // of while
C
Tab Width: 8
Ln 32, Col 21
INS
4 : 3
this will show up FIFTEEN times
5 : 5
this will show up FIFTEEN times
6 : 8
this will show up FIFTEEN times
7 : 13
this will show up FIFTEEN times
8 : 21
this will show up FIFTEEN times
9 : 34
this will show up FIFTEEN times
10 : 55
this will show up FIFTEEN times
11 : 89
this will show up FIFTEEN times
12 : 144
this will show up FIFTEEN times
13 : 233
this will show up FIFTEEN times
14 : 377
dkuntz@lap118a55:~/cs370/lab9> ./lab9 -o small < test5.decaf
dkuntz@lap118a55:~/cs370/lab9> gcc small.s
dkuntz@lap118a55:~/cs370/lab9> ./a.out
404142can handle arrays too...
0 : 0
can handle arrays too...
1 : 1
can handle arrays too...
2 : 1
can handle arrays too...
3 : 2
can handle arrays too...
4 : 3
can handle arrays too...
5 : 5
can handle arrays too...
6 : 8
can handle arrays too...
7 : 13
can handle arrays too...
8 : 21
can handle arrays too...
9 : 34
can handle arrays too...
10 : 55
can handle arrays too...
11 : 89
can handle arrays too...
12 : 144
can handle arrays too...
13 : 233
can handle arrays too...
14 : 377
dkuntz@lap118a55:~/cs370/lab9>