Control-Flow in Programs
Examples of Control-Flow Artefacts

Instructor: Firstname Lastname, Anonymous Institute, Country
firstname.lastname@institute.email

Month DD, YYYY

Example Question and Answer

Program

```
0
  a = read()
1
  a = a + 1
2
  # end
```

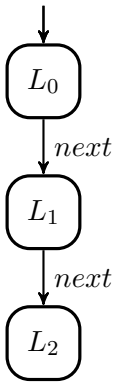
Structural Abstraction

```
0
  expression assignment
1
  expression assignment
2
  # end
```

Table of Control Transfer Functions

i	next	error
0	1	2
1	2	2
2		

Control Flow Graph



Execution path for the input `a = 1`.

$$L_0 \xrightarrow{next} L_1 \xrightarrow{next} L_2$$

Section 1

Program

```
0
  x = read()
1
  y = 1
2
  z = x + y
3
  # end
```

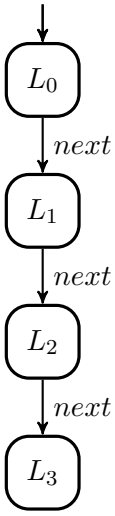
Write the structural abstraction of the program.

```
0
  expression assignment
1
  expression assignment
2
  expression assignment
3
  # end
```

Fill in the table of Control Transfer Functions with the appropriate locations for the program.

i	next	error
0	1	3
1	2	3
2	3	3
3		

Draw the Control Flow Graph (CFG) for the program.



Trace the actual execution of the program for the input `x = 2`.

$L_0 \xrightarrow{next} L_1 \xrightarrow{next} L_2 \xrightarrow{next} L_3$

Section 2

Program

```
0
  x = read()
1
  if x > 10:
2
      y = 2
3
  else:
4
      y = x
5
      x = x * 2
6
  z = x - y
7
  # end
```

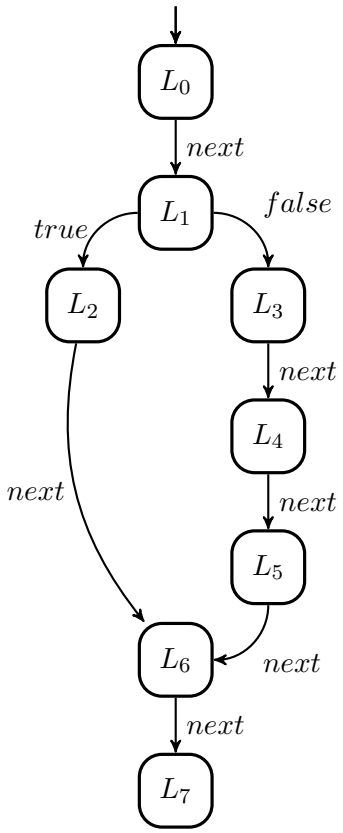
Write the structural abstraction of the program.

```
0
  expression assignment
1
  if:
2
      expression assignment
3
  else:
4
      expression assignment
5
      expression assignment
6
  expression assignment
7
  # end
```

Fill in the table of Control Transfer Functions with the appropriate locations for the program.

i	next	true	false	error
0	1			7
1		2	3	7
2	6			7
3	4			
4	5			7
5	6			7
6	7			7
7				

Draw the Control Flow Graph (CFG) for the program.



Trace the actual execution of the program for the input `x = 6`.

$$L_0 \xrightarrow{next} L_1 \xrightarrow{false} L_3 \xrightarrow{next} L_4 \xrightarrow{next} L_5 \xrightarrow{next} L_6 \xrightarrow{next} L_7$$

Section 3

Program

```
0
  i = read()
1
  x = 5
2
  while i > 0:
3
      i = i - 1
4
      y = x * i
5
      continue
6
  x = i + y
7
  # end
```

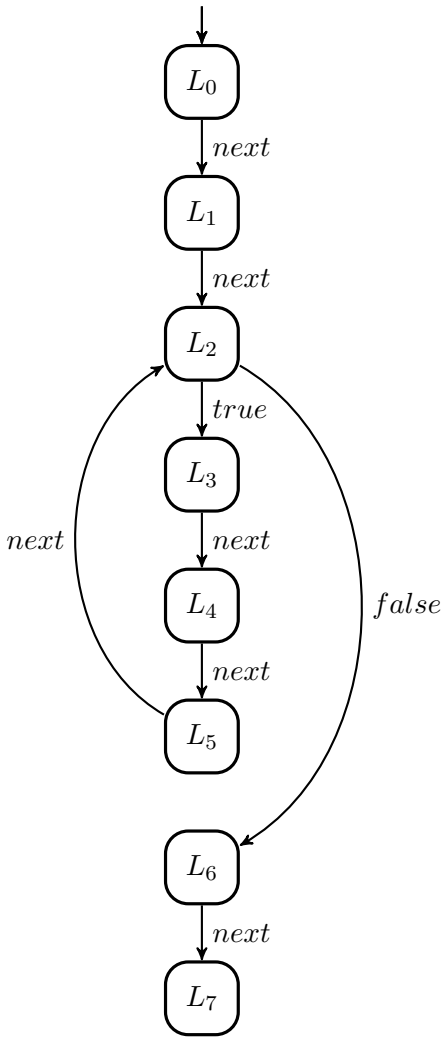
Write the structural abstraction of the program.

```
0
  expression assignment
1
  expression assignment
2
  while:
3
      expression assignment
4
      expression assignment
5
      continue
6
  expression assignment
7
  # end
```

Fill in the table of Control Transfer Functions with the appropriate locations for the program.

i	next	true	false	error
0	1			7
1	2			7
2		3	6	7
3	4			7
4	5			7
5	2			
6	7			7
7				

Draw the Control Flow Graph (CFG) for the program.



Trace the actual execution of the program for the input `i = 2`.

$$L_0 \xrightarrow{next} L_1 \xrightarrow{next} L_2 \xrightarrow{true} L_3 \xrightarrow{next} L_4 \xrightarrow{next} L_5 \xrightarrow{next} L_2 \xrightarrow{true} L_3$$
$$\xrightarrow{next} L_4 \xrightarrow{next} L_5 \xrightarrow{next} L_2 \xrightarrow{false} L_6 \xrightarrow{next} L_7$$

Section 4

Program

```
0
  def add(a, b):
1
    c = a + b
2
    return c
3
  def modify(x):
4
    x =  x * 2
5
    k = add(x, 1)
6
    return k
7
  m = add(5, 3)
8
  n = modify(m)
9
  # end
```

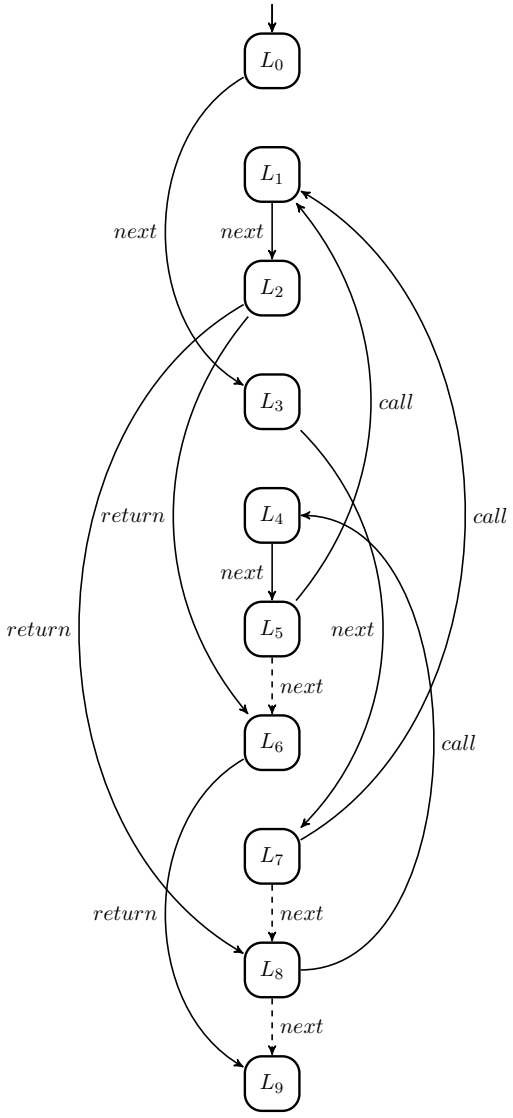
Write the structural abstraction of the program.

```
0
  def add:
1
      expression assignment
2
      return
3
  def modify:
4
      expression assignment
5
      call add assignment
6
      return
7
  call add assignment
8
  call modify assignment
9
  # end
```

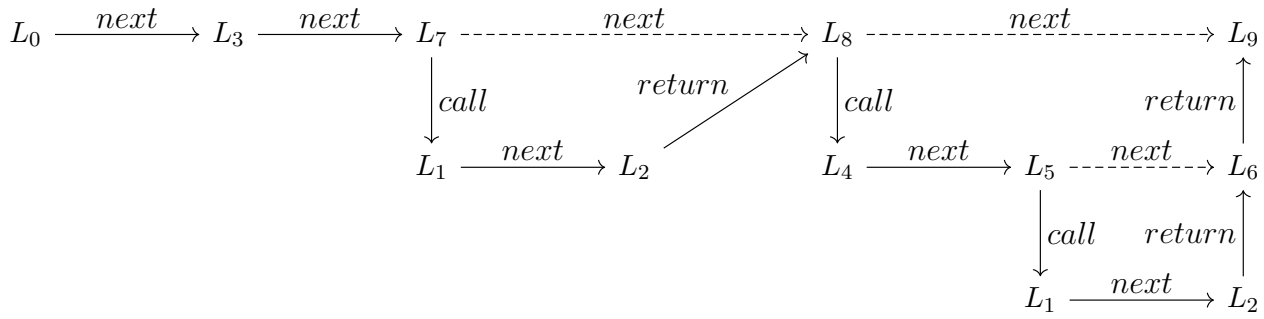
Fill in the table of Control Transfer Functions with the appropriate locations for the program.

i	next	call	ret	error
0	3			
1	2			9
2			{8, 6}	9
3	7			
4	5			9
5	6	1		9
6			{9}	9
7	8	1		9
8	9	4		9
9				

Draw the Control Flow Graph (CFG) for the program.



Trace the actual execution of the program.



Section 5

Program

```
0
  def func(x):
1
    y = 5//x
2
    return x - 1
3
  i = read()
4
  while i >= 0:
5
    i = func(i)
6
    continue
7
  # end
```

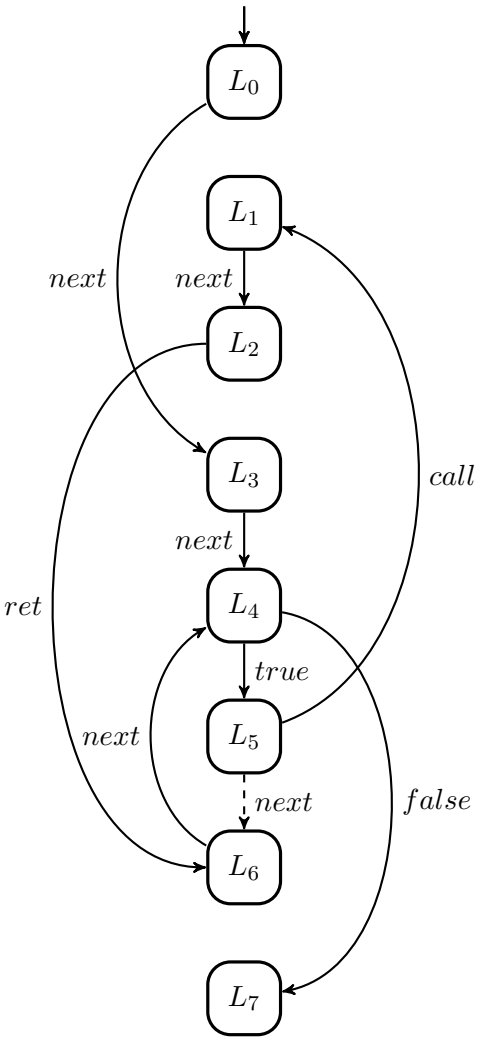
Write the structural abstraction of the program.

```
0
  def func:
1
      expression assignment
2
      return
3
  expression assignment
4
  while:
5
      call func assignment
6
      continue
7
  # end
```

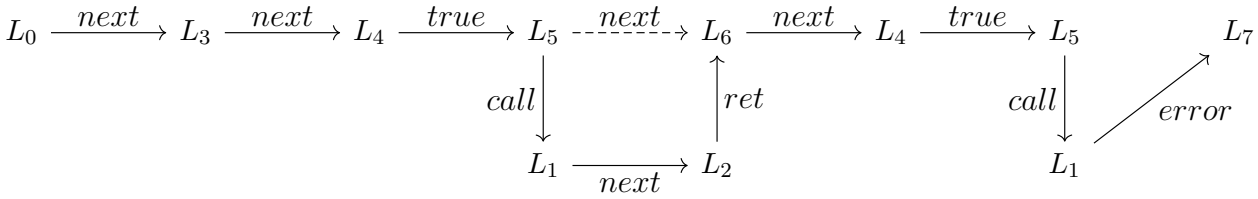
Fill in the table of Control Transfer Functions with the appropriate locations for the program.

i	next	true	false	call	ret	error
0	3					
1	2					7
2					{6}	7
3	4					7
4		5	7			7
5	6			1		7
6	4					
7						

Draw the Control Flow Graph (CFG) for the program.



Trace the actual execution of the program for the input `i = 1`.



Space for Rough Work