

C to LC3 Examples...What C code generated the assembly code shown...

Symbol Table:

Identifier	Type	Offset	Scope
A	int	0	Global
B	int	2	Global
X	int	0	main
Y	int	-1	main
Z	int	-2	main

Code 1:

```
AND R0, R0, #0
ADD R0, R0, #5
STR R0, R5, #0
```

Code 2

Initialize R0 to 5 and store this value into R5 + offset 0 which is address of local variable X
X= 5;

```
AND R1, R1, #0
ADD R1, R1, #3
STR R1, R4, #0
```

Code 3:

Initialize R1 to 3 and store into address R4 + offset 0..
R4 is global pointer, so this accesses global variable A
A =3;

```
LDR R1, R5, #0
ADD R1, R1, #5
STR R1, R5, # -1
```

Code 5:

Load value of local variable X into register R1, and then add 5, and store this in memory address R5 + (-1) which is local var Y
Y = X +5;

```
LDR R0, R4, #2
STR R0, R5, # -2
```

Code 6:

Load value of global var B into R0 and store this into local var Z
Z = B;

```
ADD R0, R5, # -1
STR R0, R5, # -2
```

Set R0 equal to R5 + (-1) which is address of local var X
store this value (the address of X) to local variable Z
Z = &X;

C to LC3 Arrays: Examine the code segments below. What are the LC3 instructions generated by a C compiler for the C code below. The symbol table is shown below. grid is array of size 10.

Symbol Table

Identifier	Type	Offset	Scope	
grid	int	-9	foo	
x	int	-10	foo	
ptr	int	-11	foo	
i	int	-12	foo	

Recall: R5 is frame pointer (dynamic link), R6 is top of stack, R4 is pointer to static area.

```
grid[6] = 5;
```

```
AND R0, R0, #0 ; set value of 5 in a register
```

```
ADD R0, R0, #5
```

```
ADD R1, R5, #-9; get address of grid[0]
```

```
ADD R1, R1, #6; add 6 to that address
```

```
STR R0, R1, #0; store 5 into that address
```

```
x= grid[3] +1;
```

```
ADD R0, R5, # -9; get address of grid[0]
```

```
ADD R0, R0, #3 ; add 3 to the address
```

```
LDR R1, R0, #0 ; fetch value at grid[3]
```

```
ADD R1, R1, #1 ; add 1 to it
```

```
STR R1, R5, # -10; store into x
```

```
grid[i] = x;
```

```
LDR R0, R5, #-10 ; get value of x into a register
```

```
LDR R1, R5, #-12 ; get value of i into register
```

```
ADD R2, R5, # -9 ; get address grid[0]into register
```

```
ADD R2, R2, R1 ; and add i to it to get address of grid [i]
```

```
STR R0, R2, #0 ; store value of x into this address
```

```
ptr = grid;
```

```
ADD R0, R5, # -9 ;get address of grid[0]
```

```
STR R0, R5, #-11 ; store this value into ptr
```

C to LC3 Pointers: Examine the code segments below. What are the LC3 instructions generated by a C compiler for the C code below. The symbol table is shown below...

Symbol Table

Identifier	Type	Offset	Scope	
i	Int	0	main	
ptr	Int	-1	main	

Recall: R5 is frame pointer (dynamic link), R6 is top of stack, R4 is pointer to static area.

i=4;

```
AND R0, R0, #0
ADD R0, R0, #5
STR R0, R5, #0
```

ptr = &i;

```
ADD R0, R5, #0      ; get address of i
STR R0, R5, #-1     ; store in location for ptr
                    ; so that local var ptr
                    ; contains address of i
```

***ptr = *ptr +1;**

```
LDR R0, R5, #-1     ; get address stored at ptr
LDR R1, R0, #0      ; dereference ptr to get value
ADD R1, R1, #1      ; add one to the value
STR R1, R0, #0      ; store result where the pointer points
```