

CS 2461 - Quiz 8: You can discuss the questions at your table.

NAME: _____

Examine the code segments below. What are the LC3 instructions generated by a C compiler for the C code below (assume the instructions are in function foo). The symbol table is shown below. grid is array of size 10.

```
int foo(int a){
    int grid[10];
    int x=10;
    int *ptr;
    int i;
    ...}
```

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/global area.

```
ptr = &x; /* set ptr to point to grid */
```

```
ADD R0, R5, #-10; get address of grid[0] into R0
                    ;this is the value of ptr (an address)
STR R0, R5, #-11 ; so store R0 in value of ptr (offset 011)
```

```
*ptr = *ptr +1; /* increment value that ptr points to */
```

```
;dereference ptr to get the value it is pointing to and store into R0
LDR R0, R5, # -11 ; get address ptr is pointing to
LDR R1, R0, #0 ; load value (dereference ptr)
ADD R1, R1, #1 ; add 1 to this value and
STR R1, R0, #0 ; store this value at address ptr is pointing to
```

```
grid[i] = grid[i] + x; /* increment value of grid[i] by x */
```

```
; first get value of i into R0, get address of grid[0] into R1 and add
value of i to R1 to get address of grid[i]
LDR R0, R5, # -12 ; load value of i into R0
ADD R1, R5, #-9 ;set R1 = address of grid[0]
ADD R1, R1, R0 ; set R1= address of grid[i]
LDR R2, R1, #0 ;load value of grid[i] into R2, i.e., R2= grid[i]
LDR R3, R5, #-10 ; load x into R3, i.e, R3= x
ADD R2, R2, R3 ;set R2= grid[i] + x and store this into grid[i]
STR R2, R1, #0 ; store into grid[i]
```

More examples...

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

```
LDR R0, R5, # -12      ; load value of i into R0
ADD R1, R5, # -9; set R1 = address of grid[0]
ADD R1, R1, #3  ; add 3 to R1 to get address of grid[3]
LDR R2, R1, #0      ; set R2 = grid[3]
ADD R2, R2, R0  ; set R2 = grid[3] + i
STR R2, R5, #-10; store R2 to x, i.e, x = grid[3] + i
```

```
i = *ptr;
```

```
LDR R0, R5, #-11; get value of ptr and dereference it
LDR R1, R0, #0      ; R1 = *ptr
STR R1, R5, # -12  ; store into i
```

```
*ptr = x;
```

```
LDR R0, R5, #-10; get value of x into R0
LDR R1, R5, #-11; get value of (address in) ptr
STR R0, R1, #0      ; store into this address to set *ptr = x
```

```
ptr = &i;
```

```
ADD R0, R5, #-12 ; get address of i and put in register R0
STR R0, R5, #-11 ; store this as value of ptr
```