

C to/from LC3

CIT 593

Example 1

```
int main()  
{  
  int a = 0;  
  int b = 5;  
  int b = 2;  
  int d = 1;  
  a = b + c + d;  
}
```

Assume that, a, b, c, d
are initialized for you

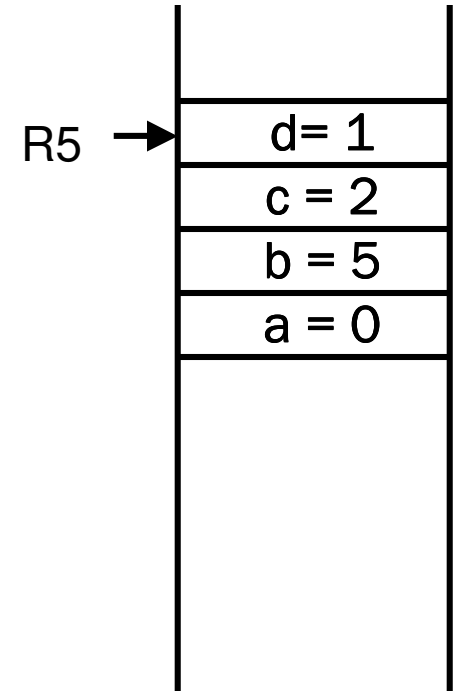
Write the LC3 code for
 $a = b + c + d$

Answer on next slide.

Solution Example 1

$a = b + c + d$

```
LDR R1, R5, #2 ; R1 = b  
LDR R2, R5, #1 ; R2 = c;  
LDR R3, R5, #0 ; R3 = d  
ADD R1, R1, R2 ; R1 = b + c  
ADD R1, R1, R3 ; R1 = R1 + d  
STR R1, R5, #3 ; Store a
```



Local variables get put onto stack in the order in which they appear. Since d is last, it goes on top of the stack

Example 2

...

```
int a[10]
```

```
int x = 5;
```

....

Also assume that array 'a' is initialized

Write the LC3 equivalent code for

```
a[x] = a[x + 1]
```

Answer on next slide.

Solution Example 2

- $a[x] = a[x + 1]$

LDR R0, R5, #0 ; R0 = x

ADD R0, R0, #1 ; R0 = x+1

ADD R1, R5, #1 ; R1 = &a[0]

ADD R1, R0, R1 ; R1 = &a[x+1]

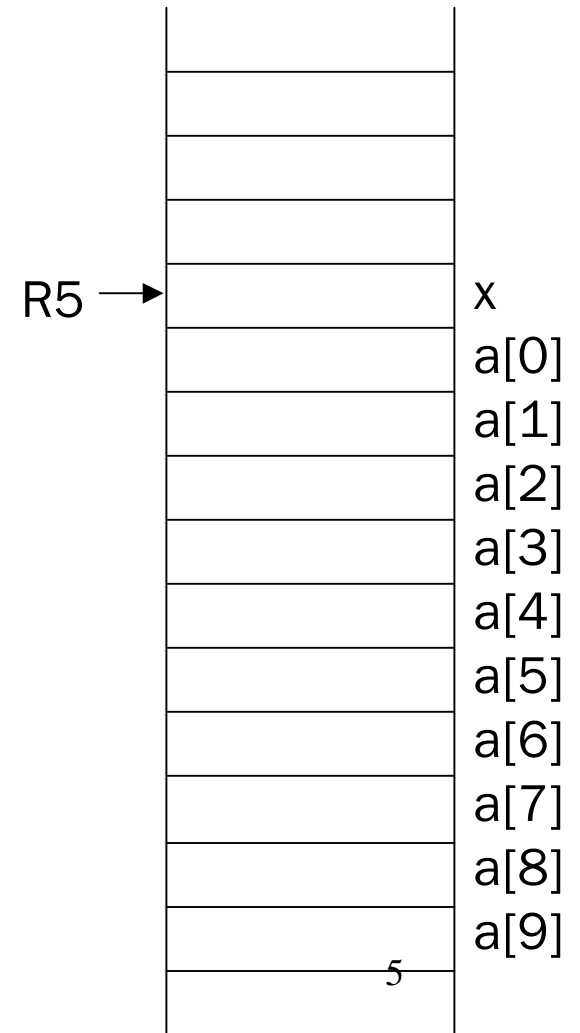
LDR R2, R1, #0 ; R2 = a[x + 1]

LDR R0, R5, #0 ; R0 = x

ADD R1, R5, #1 ; R1 = &a[0]

ADD R1, R0, R1 ; R1 = &a[x]

STR R2, R1, #0 ; a[x] = R2 = a[x+1]



Example 3

```
struct temp{
    int lowT;
    int highT;
};
typedef struct temp T;

int main( ){
    T day;
    int avg;
    day.lowT = -1; ....(1)
    day.highT = 15; ....(2)
    ...
    .....
    return 0;
}
```

Write LC3 code for C
code shown in lines (1)
and (2)

Answer on next slide.

Solution to Example 3

- `day.lowT = -1 & day.highT = 15`

```
AND R0, R0, #0
```

```
ADD R0, R0, #-1
```

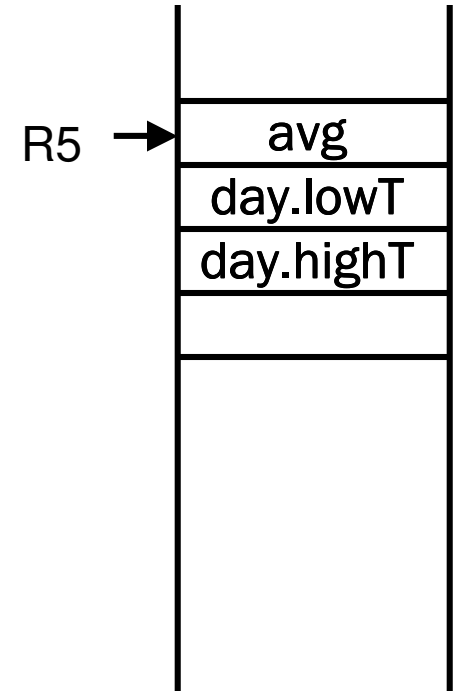
```
AND R1, R1, #0
```

```
ADD R1, R1, #15
```

```
ADD R2, R5, #1 ;R2 = start of struct
```

```
STR R0, R2, #0 ;day.lowT = R0
```

```
STR R1, R2, #1 ;day.highT = R1
```



Example 4

In main() , the local variables are defined in the following order

```
int a;
```

```
int * b;
```

```
int c ;
```

- Below is the LC3 code

```
ADD R0,R5, #2
```

```
STR R0, R5, #1
```

```
LDR R1, R5, #1
```

```
LDR R1, R1, #0
```

```
ADD R1, R1, #1
```

```
STR R1, R5, #0
```

The LC3 code is doing something with a, b, c.

Translate the LC3 code to C

Answer on next slide.

Solution to Example 4

Equivalent C code

```
b = &a;
```

```
c = *b + 1;
```

```
ADD R0,R5, #2 ; R0 = &a = addr of a
```

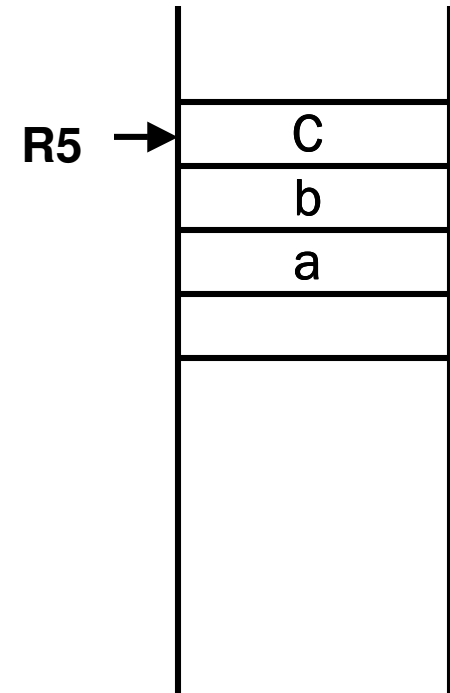
```
STR R0, R5, #1 ; b = &a
```

```
LDR R1, R5, #1 ; R1 = &a
```

```
LDR R1, R1, #0 ; R1 = *b i.e. value of a
```

```
ADD R1, R1, #1 ; R1 = *b + 1
```

```
STR R1, R5, #0 ; c = R1
```



Example 5

The variables are declared in the following order:

```
int i;
```

```
int a[10];
```

```
    AND R0, R0, #0
    STR R0, R5, #10
    ADD R0, R5, #0
LOOP  LDR R1, R5, #10
    AND R2, R2, #0
    ADD R2, R1, # -10
    BRz  DONE
    STR R1, R0, #0
    ADD R0, R0, #1
    ADD R1, R1, #1
    STR R1, R5, #10
    BRnzp LOOP
DONE  ...
```

Translate the LC3 code to C

Answer on next slide.

Solution to Example 5

- i is at location $R5 + 10$
- $a[0]$ is at location $R5 + 0$
- $a[9]$ is at location $R5 + 9$

- C code:

```
for(i = 0; i < 10; i++){  
    a[i] = i;  
}
```