CS1022 Tutorial #1
Addressing Modes and Arrays

(a) Describe precisely the operation performed by each of the instructions below. For each instruction, state the memory address that is accessed and the new value contained in the base address register, if it has been modified.

```
1    LDR     R0, [R1, #8]
2    LDRB    R0, [R1, #1]!
3    LDR     R0, [R1], #4
4    STR     R0, [R1, R2, LSL #2]
```

Assume the following initial values in R1 and R2:

`R1=0xA1000080, R2=0x00000042`

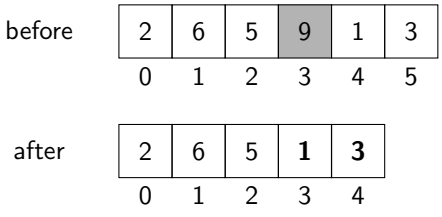
(b) Translate each of the pseudo-code statements below into ARM Assembly Language. Assume that variables f, c and r correspond to registers R4, R5 and R6 respectively. A is a one-dimensional array of 16-bit unsigned values with a starting address contained in R10. B is a 16×16 (two-dimensional) array of 32-bit unsigned values with a starting address contained in R11. (Assume that B is stored in row-major order.)

(i) f = A[c];

(ii) A[r] = A[r+2];

(iii) f = B[c][r] * B[r][c]

(c) Write an ARM Assembly Language program that will remove an array element from a specified index in an array of word-size values. The figure below illustrates an array in which an element is removed from index 3. **When removing the element from the array, your program should move the subsequent elements towards the start of the array to fill the "gap" created in memory.** Begin by storing the start address of a test array in R0 and the index of a test element to remove in R1. Assume the number of elements in the array is stored in R2.



(d) Write an ARM Assembly Language program that will insert a new array element into a specified index in an array of word-size values. The figure below illustrates an array in which an element is inserted at index 2. **When inserting the element into the array, your program should move the subsequent elements towards the end of the array to create a "gap" in memory to contain the new element.** (Assume that sufficient memory has been allocated to allow you to extend the length of the array as required.) Begin by storing the start address of a test array in R0, the index of a test element to insert in R1 and the value of the new element in R2. Assume the number of elements in the array is stored in R3.