Additional Midterm 2 Practice Questions

Programming Questions:

1. For this problem, you will write a C function that reverses the elements in an array. The function's two parameters are the array and its size (total number of elements).

For example, if **`A[0]`** = 4, **`A[1]`** = 5, and **`A[2]`** = 6 before the call to **`ReverseArray`**, then after the function call **`A[0]`** = 6, **`A[1]`** = 5, **`A[2]`** = 4.

```
int ReverseArray(int array[], int arraySize)
{




}
```

2. In mathematics, the dot product of any two vectors

vectorA = [a_0 a_1 … a_{n-2} a_{n-1}]
vectorB = [b_0 b_1 … b_{n-2} b_{n-1}]

is defined by

vectorA o vectorB = (a_0 * b_0) + (a_1 * b_1) + … + (a_{n-2} * b_{n-2}) + (a_{n-1} * b_{n-1})

Complete the following function in C to calculate the dot product of two vectors. Here, the vectors are defined by the two integers arrays `vectorA` and `vectorB`. Their length is `vectorLength`.

```
int DotProduct (int vectorA[], int vectorB[], int vectorLength)
{









}
```

3. Given an array with just 0's and 1's find the length of the largest subarray with equal number of 0's and 1's. For example, given an array

0 1 1 1 1 0 0 0 1

the largest subarray with equal number of 0's and 1's has the length 8, from index 0 to index 7.

Write a **recursive** function largestSubarray. This function takes 3 inputs: an array - A, the index of its first element – start, the index of its last element – end, and returns the size of the largestSubarray. If no subarray with equal number of 0's and 1's is found, the function should return 0.

```
int largestSubarray(int * A, int start, int end){




}
```

4. For this problem, find the number of non-overlapping pairs that appear in a positive integer array **using recursion**.

For example:
{11, 5, 7, 9, 11, 3, 5} has a total of 2 pairs: 1 pair of 11 and 1 pair of 5
{11, 5, 7, 9, 11, 11} has a total of 1 pair: a pair of 11
{11, 5, 7, 9, 11, 3, 11, 5, 11} has a total 3 pairs: 2 pairs of 11 and 1 pair of 5

Write a **recursive function** findpair. This function takes 3 arguments: the positive integer array, the start index of the array and the end index of the array. It returns the total count of pairs found in this array. If no pair is found, this function should return 0. **During the search process, you are allowed to modify elements in the array.**
Hint:

- Think about what would be the base (terminal) case
- Think about what would be the recursive (reduction) case

```
int findpair(int array[], int start, int end){

}
```

5. You will provide the data values that appear on the run-time stack during the execution of simple C program. Part of the stack frame for function main is shown in the memory table in Part A and Part b. R6 is the stack pointer and R5 is the frame pointer.

Part A: In the memory table, draw the stack right before "z = foo(x, n);" is called. **Also indicate the values of R5 and R6 at this point of program execution.**

Part B: In the memory table, draw the stack right after "z = foo(x, n);" is called and before control is transferred to foo. **Also indicate the values of R5 and R6 at this point of program execution.**

Part C: Convert the foo function from C to an LC-3 subroutine with correct use of the run-time stack.

```
int foo(int a[], int b)
{
     int c=0, d=0;

     while(d < b)
     {

          c += a[d];
          d++;
     }

     return c;
}

int main()
{
     int n = 3;
     int x[n] = {1,3,5};

     z = foo(x, n);

     return 0;
}
```


Part A:

xBCC8	
xBCC9	
xBCCA	
xBCCB	
xBCCC	
xBCCD	
xBCCE	
xBCCF	
xBCD0	
xBCD1	
xBCD2	
xBCD3	
xBCD4	
xBCD5	
xBCD6	
xBCD7	
xBCD8	x[ ]= ____
xBCD9	x[ ]= ____
xBCDA	x[ ]= ____
xBCDB	n = 3

R5 = _____ ; R6 = _____

Part B:

xBCC8	
xBCC9	
xBCCA	
xBCCB	
xBCCC	
xBCCD	
xBCCE	
xBCCF	
xBCD0	
xBCD1	
xBCD2	
xBCD3	
xBCD4	
xBCD5	
xBCD6	
xBCD7	
xBCD8	x[ ]= ____
xBCD9	x[ ]= ____
xBCDA	x[ ]= ____
xBCDB	n = 3

R5 = _____ ; R6 = _____

Part C:

```
;;FOO Subroutine
;callee setup – push bookkeeping info and local variables




;function logic




;callee tear-down – pop local variables, bookkeeping info
```

Concept Questions:

1. In LC-3, how many bytes of memory are needed to store a pointer of type int star (int *int_ptr)?

2. By default, global variables are stored in which region of memory?

3. What is the significance of the function main in C?

4. What condition is required to perform binary search on an array?

5. Consider the following C code, which is part of a larger C program.

```
int sum = 0;
int count = 9;
do  {
     sum = sum + count;
     count--;}
while (count > 5);

printf("sum = %d\n",sum);
```

What will be printed by the `printf` statement?

`sum =`

6. Assume j is declared earlier in the code. Execution of the `if` statement below <u>always / sometimes / never (circle one)</u> results in execution of the `printf` statement shown below. Explain your choice.

```
if (j = 0)
     printf ("j is equal to 0\n");
```