Answer all questions in a separate document and email that document to abrahamsonk@ecu.edu as an attachment. Make your answers clear, concise and precise. All questions concern the C++ programming language. **Check your answers.**

1. If variable $p$ has type Fruit*, what is the type of C++ expression $*p$?

2. If $x$ has type Fruit and $p$ has type Fruit*, is statement

   $$p = x;$$

   allowed? (Yes or no. It is allowed if it will compile and will do something sensible when performed by the program.)

3. Write a statement that creates variable $A$ and makes $A$ point to a newly allocated array of 20 integers in the heap.

4. Suppose that variable $A$ has been created and initialized as in question 3. Write a statement that stores 95 into the first variable in array $A$. (The first variable is the one with the smallest index.)

5. If $A$ has been created and initialized as in question 3, is statement

   $$A[20] = -1;$$

   allowed? (Yes or no. It is allowed if it will compile and will do something sensible when performed by the program.)

6. Suppose that variable $p$ has already been created and has type double*. Also suppose that $p$ has been initialized to point to newly allocated memory in the heap. Write a statement that stores 0.75 into the memory to which $p$ points.

7. Write a C++ definition of function rotate($s$) that takes a null-terminated string $s$ and yields a null-terminated string that you get by moving the first character of $s$ to the end. For example, rotate("rabbit") = "abbitr" and rotate("frog") = "rogf". The result string must be allocated in the heap. Function rotate($s$) cannot change what is in array $s$.

   You can use functions from the cstring library. A heading is given.

   ```cpp
   char* rotate(const char* s)
   ```
8. What are the values of variables $a$, $b$, $c$, $x$ and $y$ after performing the following sequence of statements? (Hint. Work out the answer carefully, and show your work. Draw pointer diagrams.)

In order to receive any credit for this problem, you must get at least 3 of the 5 answers correct.

```c
int a, b, c;
int x = 25;
int y = 62;
int* q = &x;
int* p = &y;
int* r = q;

*r = 100;
*p = *q;
*q = 78;
r = p;
*p = 9;
p = q;
q = r;
a = *p;
b = *q;
c = *r;
```


Write a function `reverse(A, m, n)` that reverses the order of $A[m], A[m+1], \ldots, A[n-1]$, so that those values end up in descending order. Assume that $0 \leq m \leq n \leq s$. If $m = n$, then there are no values to be reversed.

Do not allocate a new array. Make the changes in array $A$.

**Use recursion for this problem. Do not use any kind of loop.** A heading is given.

You can use function `swap(x, y)`, which swaps the contents of variables $x$ and $y$.

(Hint. Look at small examples on paper. What does `reverse(A, m, n)` need to do if $m \geq n + 1$? What if $m < n + 1$?)

```c
void reverse(int A[], int m, int n)
```