Name: _____

Midterm Exam (20 points)

Please submit only CPP files that you write including test programs (e.g. main()) that show how you test these functions. No output or screenshot is needed.

Important requirements for all questions:

- No global variable is allowed to be declared and used. Constants are ok.
- Functions cannot use "cin" or "cout" unless it is explicitly permitted. It should make use of parameters and return value instead. "cin" and "cout" should be done in main() or any testing functions
- Multiple return statements in one function are not allowed.
- Each question is worth of 5 points

1. Write a function named "**countAB**" that accepts a C-string (an array of characters terminating by a NULL character. It returns the count of how many "AB" in that array. In addition, it also return a boolean indicating whether the given array starts with "AB" in the beginning of the array.

   For example, these are the C-strings with their expected return values (counter and flag)

   "AB" 1 and true
   "ABCABC" 2 and true
   "ABCABCABC" 3 and true
   "AAB" 1 and false
   "AABCABC" 2 and false
   "" 0 and false
   "A" 0 and false
   "BA" 0 and false
   "ACB" 0 and false
   "AACB" 0 and false

   Note: this function cannot use the string class or string functions such as strlen. It should only use an array of characters with a null terminating character (C-string)

2. Write a function named "**hasTheMaxInTheBeginningOrEnd"** that accepts an array of integers and its size. It will go through the array and return true if the maximum number in the array is the first or last in the array. You can assume the array having at least one number. In addition, it also returns the actual value of the maximum value.

   For example, the following arrays and their expected return values:

   ```
   int numList0[] = {10} ;                 // true and 10
   int numList1[] = {10, 20, 30, 40} ;     // true and 40
   int numList2[] = {40, 10, 30, 50} ;     // true and 50
   int numList3[] = {10, 40} ;             // true and 40
   int numList4[] = {40, 10} ;             // true and 40
   int numList5[] = {30, 20, 10} ;         // true and 30
   ```

```
int numList6[] = {10, 20, 40, 30} ;          // false and 40
int numList7[] = {20, 30, 10} ;              // false and 30
int numList8[] = {30, 35, 1, 20} ;           // false and 30
```

3. Write a function named "**readNumUntilFirstNumRepeated**" that reads in a list of numbers until the user enters the first number again. It will return the total count of numbers the user has entered and the value of the first number. Please note that first number should be counted only once.

   Mote: this function will use cout to prompt and cin to read in the numbers but it will return the values to the caller.

   Here is an example of a test call:

```
int value = 0 ;
cout << "Count: " << readNumUntilFirstNumRepeated(value) << endl ;
cout << "Value of the first number: " << value << endl;
```

   and its expected input/output from several separate runs:

```
Please enter numbers until the first is repeated:
10
20
30
40
10
Count: 4
Value of the first number: 10

Please enter numbers until the first is repeated:
10
10
Count: 1
Value of the first number: 10

Please enter numbers until the first is repeated:
50
10
50
Count: 2
Value of the first number: 50
```

4. Write a function named "**payOut**" that accepts an array of working hours, its size and another same size array of double for the wages per hour.
   The relationship between these two arrays is the index.

It will calculate the total payout by going through the array of working hours and multiply with its corresponding wages per hour. Please note that it will calculate overtime pay of 1.5 whenever the working hour is more than 40.

It will return the total payout for all and the number of overtime.

Note: the function should traverse the array only once.
The function should not use cin or cout but return the values to the caller.

For example,

```
int  hours1[] = {30, 40, 50} ;
double  wages1[]  = {10.0, 20.0, 30.0} ;
```

It will return the total pay of $2750 and 10 hour overtime.
(30 * 10.0) + (40 * 20.0) + (40 * 30.0) + (10 * 45.0) = 2750

And the arrays

```
int  hours2[] = {1, 41} ;
double  wages2[]  = {10.0, 20.0} ;
```

It will return the total pay of $840 and 1 hour overtime.
(1 * 10.0) + (40 * 20.0) + (1 * 30.0) = 840