**Northeastern Illinois University**

**CS-200: Programming I**

**PLTL: Arrays & Loops**

**Problem 1**

* Write a program that has the class name Problem1 and that has the main method.
* Prompt the user to enter a positive integer n. You can assume that the user will enter the

correct value.

* Then, prompt the user to enter n integers.
* Determine the minimum number entered and print out all of the integers that were entered **after** the minimum number. If no integers were entered after the minimum, print out "No numbers after min". If there are repeats for the minimum, use the first occurrence of the value.
* Several sample runs are provided for you below. Your output must be formatted **exactly**

like the sample runs below.

|  |
| --- |
| Enter n (> 0): 5  Enter 5 integers: 8 2 -5 -3 10  After minimum: -3 10 |

|  |
| --- |
| Enter n (> 0): 3  Enter 3 integers: 0 4 -1  No numbers after min |

|  |
| --- |
| Enter n (> 0): 6  Enter 6 integers: -3 -3 6 10 -1 0  After minimum: -3 6 10 -1 0 |

|  |
| --- |
| Enter n (> 0): 1  Enter 1 integer: 99  No numbers after min |

**Problem 2**

* Write a program that has the class name Problem2 and that has the main method. Leave

the main method empties for now.

* Write a method named doubleChar that takes one parameter, a character array a and

returns a new character array.

* The method should create a new character array that has two (i.e. doubles) of every

element in a and return that array.

* Create a printArray method that takes a character array as a parameter and prints out

the elements of the array on the same line separated by a space.

* Several sample usages are provided for you below. Use the sample usages in the main

method to test your code (and use the printArray method to print out the results of

calling the doubleChar method!).

|  |  |
| --- | --- |
| Sample Method Usage | Return Value |
| **char** [] a1 = {‘a', 'A', 'j', 'h'};  **char** [] b1 = doubleChar(a1); | {'a', 'a', 'A', 'A', 'j', 'j', 'h', 'h'} |
| **char** [] a2 = {'b', 'b', 'c'};  **char** [] b2 = doubleChar(a2); | {'b', 'b', 'b', 'b', 'c', 'c'} |

**Problem 3**

* Write a program that has the class name Problem3 and that has the main method.
* Prompt the user to enter a positive integer n greater than 1. You can assume that the user

will enter a positive value.

* Then prompt the user to enter n integers.
* Move all the 0’s to the end of an array. Maintain the relative position of the other (nonzero)array elements and print out the new array.
* You should also print out the number of 0’s as well as the number of integers that are

before 0. If there are no 0’s, then print "There are no zeros in the array."

* Several sample runs are provided for you below. Your output must be formatted **exactly**

like the sample runs below. Use the sample usages in the main method to test your code.

|  |
| --- |
| Enter n (> 1): 8  Enter 8 integers followed by the space: 0 0 3 0 2 7 0 9  The new array with 0's at the end: 3 2 7 9 0 0 0 0  There are 4 zeros in the array and there are 4 integers before them. |

|  |
| --- |
| Enter n (> 1): 7  Enter 7 integers: 4 0 3 0 0 5 8  The new array with 0's at the end: 4 3 5 8 0 0 0  There are 3 zeros in the array and there are 4 integers before them. |

|  |
| --- |
| Enter n (> 1): 4  Enter 4 integers: 22 6 89 4  There are no zeros in the array. |

**Problem 4**

* Write a program that has the class name Problem4 and that has the main method. Leave

the main method empties for now.

* Write a method named palindromeChar that takes one parameter, a character array a, and

returns a boolean.

* The program determines if a is a palindromic array. A palindromic array is an array

where the characters read the same going from the beginning to the end of the array and

reading from the end of the array to the beginning. If a is a palindromic array, then the

method should return true and false otherwise.

* For example, an array of 'r', 'a', 'd', 'a', 'r', reads same from the beginning and from the

end, so it returns true and prints out It is a palindrome. Similarly, 'w', 'a', 't', 'e', 'r', is

not a palindrome and thus returns false and prints It is not a palindrome.

* Several sample usages are provided for you below. Use the sample usages in the main

method to test your code.

|  |  |
| --- | --- |
| Sample Method Usage | Return Value |
| **char** [] a = {'r','a','c','e','c','a','r'};  **boolean** a1 = palindromeChar(a); | true |
| **char** [] b = {'w','a','t','e','r'};  **boolean** b1 = palindromeChar(b); | false |
| **char** [] c = {'f','o','o','f','a','a','r'};  **boolean** c1 = palindromeChar(c); | false |
| **char** [] d = {'c','b','a','a','b','c'};  **boolean** d1 = palindromeChar(d); | true |

**Problem 5**

* Write a program that has the class name Problem5 and that has the main method.
* Prompt the user to enter a positive integer n between 0 - 9999 until there is a repeat entry.
* The program should figure out how many integers were entered before there was a repeat entry.
* You may only use a boolean array to solve this problem.
* Several sample runs are provided for you below. Your output must be formatted **exactly**

like the sample runs below.

|  |
| --- |
| Enter an int: 87  Enter an int: 90  Enter an int: 87  You have entered 2 values |

|  |
| --- |
| Enter an int: 1  Enter an int: 1  You have entered 1 value |

|  |
| --- |
| Enter an int: 67  Enter an int: 80  Enter an int: 95  Enter an int: 30  Enter an int: 67  You have entered 4 values |

**Problem 6**

* Write a program that has the class name Problem6 and that has the main method. Within main create a char array of size 5.
* Create a string and leave it empty for now.
* Prompt the user to enter a capital grade letter (A, B, C…etc.)
* The program concatenates the grade(char) and fills out the string with a comment about the grade (example: “is a good grade!”) and prints it in one line.
* A sample usage is provided for you below.

|  |
| --- |
| Enter 5 capital letters: B A D B is a good grade! A is a good grade! D is a failing grade! |

|  |
| --- |
| Enter 5 capital letters: c d a F B c is invalid grade! d is invalid grade! a is invalid grade! F is a failing grade! B is a good grade!  |