King Abdulaziz University
Faculty of Computing and Information Technology
Computer Science Department

CPCS 202, Spring 2020

Online Final Assessment

Exam date	: 26th April, 2020 (Sunday)/3rd Ramadan, 1441H
Start Time	: 1:00pm
Estimated time needed for completion	: 2.5 - 4 hours
Total marks	: 100

Please read the following points carefully before you begin the assessment

- This assessment consists of two questions. You are required to upload two separate Java files to Blackboard from your student account as part of this assessment.
- There are no late submissions. You will not be allowed by the Blackboard system to submit after the final submission deadline ends.
- If the uploaded files are empty or you upload the wrong files, it will be solely your responsibility, and you will be awarded zero marks for that question
- Any form of cheating will result in -4. Also, please do not share your solutions with students from other sections or students from other campuses. Both the giver and the taker of the solution will be awarded -4 marks.
- Solutions which use advanced programming techniques not covered in CPCS 202 will either be awarded -4 or zero marks or as decided by the grader and based on the technique used.
- The following may be considered when assigning marks to your solutions
    - adherence to Java best practices (Java conventions) for naming variables and methods as covered in the lecture slides
    - usage of appropriate comments in code
    - usage of tabs, spaces and blank lines in code
    - appropriateness of naming variables used in code
    - Logic used in code (Remember – good programmers write simple code!!)
    - Time of submission (Early submissions are better. They demonstrate a student's good thinking ability and programming skills)
- Any updates or clarifications to the assessment will be posted on Blackboard, if required. It is your responsibility to frequently check the "*Final Assessment*" link on Blackboard for any updates.

Question 1 (40 marks)

Write a program that reads a sentence (group of words consisting of lowercase and upper case English alphabets separated by spaces) from the user and decomposes (breaks) it to characters with the following conditions

1. If the character is between *i* and *s* or between *F* and *U* it is considered special and the Unicode value for that character is printed.
2. For all other characters, the same character is printed

At the end, your program needs to display the following

1. the sum of the Unicode values for all the special characters
2. the remaining characters put together as a String

See the sample output given below for clarification. Also, note the spaces in the output when printing the Unicode values or characters in the sample output.

Note : Your solution to this question should consist of only the main method; no other method is to be created for this question.

Sample output :

```
Enter a string that has more than one word : Computer Science At KAU
C
 111
  109
   112
    u
     t
      e
       114
        83
         c
          105
           e
            110
             c
              e
               A
                t
                 75
                  A
                   85

The sum of the Unicode values of the special characters is : 904
The other characters are : CutececeAtA
```

Upload a single file named *SpecialLetters.java* to BlackBoard as your solution to this question.

Question 2 (60 marks)

You are required to write a complete program including the main method that does the following - prompts a user to enter the size of the array, creates an array of integers of the given size, then asks the user to enter numbers to fill in the array. Next, the program should display the array elements and prompt the user to enter an element to be removed. All occurrences of the element in the array should then be removed from the array.

For this, create three methods with the following description

1. A method named *findIndex* that accepts an array of integers, its size, and a number as input parameters. If the number is found in the array, the method returns the index of the element in the array. Otherwise it should return a -1 if not found.
2. A method named r*emoveElement* that accepts an array of integers, its size and an index *i* as input parameters. The method should then remove the element at index *i* from the array. After removing the element at index *i*, the remaining elements in the array should be rearranged as shown in the sample output.
3. A method *printArray* that accepts an array of integers and its size as input parameters. It should then print the array in the format shown in the output.

The *main* method should continue calling the methods *findIndex* and *removeElement* until there is no more occurrence of the element to be removed in the array.

See the sample outputs given below for clarification.

Note : The Array class and its methods should not be used to solve this question.

Sample output #1:

```
Enter the size of the array to be created: 7
Enter array elements: 12 4 32 4 4 3 16
The array has the following elements:  12 4 32 4 4 3 16
Select an element to be removed: 4
--------------------------------
Your new array is 12 32 3 16
```

Sample output #2:

```
Enter the size of the array to be created: 5
Enter array elements: 6 12 4 33 12
The array has the following elements:  6 12 4 33 12
Select an element to be removed: 33
--------------------------------
Your new array is 6 12 4 12
```

Sample output #3:

```
Enter the size of the array to be created: 5
Enter array elements: 3 5 44 21 5
The array has the following elements:  3 5 44 21 5
Select an element to be removed: 12
--------------------------------
Sorry 12 is not in the array
```

Upload a single file named *Array.java* to BlackBoard as your solution to this question.