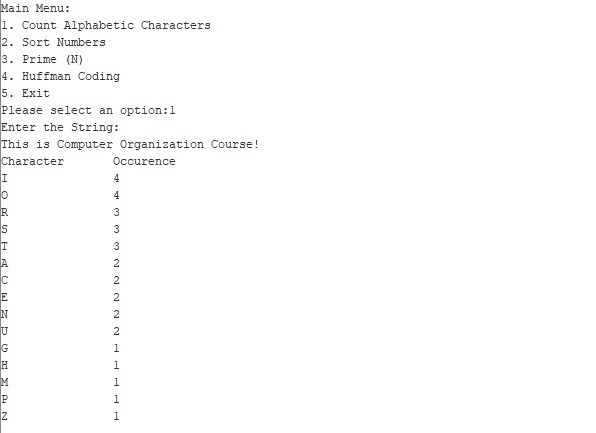
**CSE3038- COMPUTER ORGANIZATION PROJECT 1 REPORT**

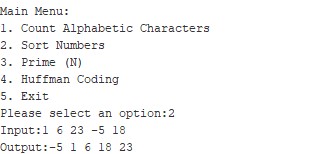
Elif Gülay Nurefşan Yücel İnsaf Özge Berktaş Student ID:150119732 Student ID:150119802 Student ID:150118072

Senanur Güvercinoğlu Student ID:150119740

**Q1:** Our approach to Q1 is to find the number of occurrences of each letter in the string entered by the user by iterating through loops. Using the range in which the ASCII values of letters correspond, we ignore the spaces, punctuations, etc., and count only the letters. While counting letters, we look at the ASCII values corresponding to the uppercase and lowercase letters separately and add them to the same array, and thus upper and lowercase letters are counted as the same character. Finally, all letters are printed in descending order according to their occurrence frequency. Equally frequent letters are also listed alphabetically among themselves.



**Q2:** Our approach for solving Q2 is to maintain one array when running the program. We start by going around all the characters of the string input with for loop, converting the characters to numbers, and adding them into the array. If the character is space then the number is added to the sequence because the number is over. If the character is a dash, since our number is negative, in another loop, the loop returns for the part after the dash, and we subtract the remaining number from 0 and add it to the array. When our array is complete, we return another for loop that sorts it out. Finally, in another for loop, the output is being printed.



**Q3:** Our approach for solving Q3 is to find the number of prime numbers by iterating through loops. At the beginning of the program, we read the integer n as an input and we initialize some registers for the loops. We created a firstLoop that initializes true for all elements of primesArray. Using a sieve of eratosthenes we try to find multiples of 2 and then find all multiples of 3, and this goes on until n. After that loop, we go to oLoop and we start counting from 2 until the square of n. If that element of primesArray is true then we go to iLoop. In the iLoop we start from the square of 2, 3, n… If the inner loop finds the multiples of that specific prime element then we initialize the element as false. This means multiple of the prime number is not prime and in countNumber we count the number of primes up to input n.

