Report Assignment II

Problem 1:

- Assumptions: It is assumed that the input integer array is less than 25 elements.
- Design Approach:
    a. In this problem a cursor is to be moved on the array and the value of this cursor is to be compared to the minimum available then maximum available, (if both turned false the number lies between min and max and their values is left as is, if one is true, its value is swapped with the cursor)
    b. To implement this method at least two min and max values, to compare the cursor to, are a must. To this end, the first and second elements of the array are compared first and accordingly set the smallest and largest to be temporary min and max respectively, this is done outside the loop since it is only done once, after that value is compared to cursor not adjacent element

Problem 2:

- Assumptions: The user-input integer is only 4 bytes not beyond that
- Design Approach:
    a. In this problem the number goes through one or two check steps depending on its nature, first the number is added with one. Since the binary representation of any odd number has to have 1 in its right most digit then when added with one, the answer is always one, if even right most is zero and result of anding is zero
    b. If number is even it is printed that it is even if it is odd, the number goes through the second check; namely whether or not it is prime. The number is halved (assuming if it had a factor it had to be between 2 and its half), the number is divided on these numbers if ever the remainder is zero then this number has a factor and hence is no prime, if the loop ends without reminder the number is prime

Problem 3:

- Assumptions: The user-input sentence is no more than 20 characters including the spaces
- Design Approach:
    a. In this problem it is obvious that the letters of the word is mirrored around the middle letter of the word, hence the size of the word is read first and a

cursor is put at the beginning while another is put at beginning + (size-1), the end.

b. The two values are compared if they are equal then the letters are mirrored as desired, if not that value of the element in the second part of the array is copied to the element in the first part, this continue until we reach the middle letter of the array by then all letters are mirrored with record of how many types changes had to be made for the mirroring to be in effect.