

PART II (45 minutes)

d. Below is an algorithm that produces palindromic numbers for most positive integers: Let n be a positive number of two digits or more $n > 9$:

- [STEP 1] If the number is palindromic STOP. If not, go to step 2
- [STEP 2] Reverse the digits
- [STEP 3] Add the reverse to the original number, the sum is your new number, go to step 1

For example:

- 39 -> 363
 - 39 is NOT palindromic
 - 39 reversed is 93
 - $93 + 39 = 132$
 - 132 is NOT palindromic
 - 132 reversed is 231
 - $231 + 132 = 363$ is palindromic after 2 iterations

[7 marks] Write a C++ function named `algo(n)` that takes a positive integer of two digits or more, $n > 9$ as an argument and returns the number of iterations required to get to a palindromic number by the procedure mentioned above. Throughout this procedure, if after 100 iterations the number is still NOT palindromic your function must return -1. You must call your functions defined in part a and b.

- `algo(11)` returns 0
- `algo(88)` returns 0
- `algo (39)` returns 2
- `algo (82)` returns 2
- `algo (59)` returns 3
- `algo (68)` returns 3
- `algo (86)` returns 3
- `algo (79)` returns 6

e. [7 marks] Write a C++ main function (see the starter code below) that allocates a dynamic array of integers of size 0 called `arr` and populates it with n (user defined) randomly generated integers from 10 to 86 by calling your function `appendElemdArr`. Then it sorts the elements of `arr` based on their number of iterations in the procedure mentioned in part (d) in ascending order. In addition, if the number of

iterations for two or more numbers are equal, your function arranges these numbers in ascending order based on their magnitudes.

Test your function by printing the elements of `arr` before and after sorting. **You must call your functions defined in part (c) and (d).**

For example, let:

`arr : {86, 88, 82, 79, 11, 59, 39, 68}`

After sorting:

`arr : {11, 88, 39, 82, 59, 68, 86, 79}`

Note that since `39` and `82` have the same number of iterations, namely 2, and `39 < 82`, the number `39` must precede `82`.

```
int main(){
    int size = 0;
    int *arr = new int[size];
    /*
        YOUR CODE
    */
}
```