

Student Name: **KEY SOLUTION** ID:

A data file named "Numbers.txt" contains positive and negative integer numbers. Write a code that reads each number from the file, if the number is positive and has more than one digit, a function **reverse()** should then take the given number as input, and return the number of digits and the reverse of the input number. The function should take the integer as input, and returns 1 if the number is positive has more than 1 digit and contains no 0 digit, and returns 0 otherwise. The test for positivity should be made inside the function. The output is saved to a new file named "result.txt" with the format shown in the sample run.

Note: Program

sample run

```
145
-54
4
91
205
432
-14
43
```

```
n = 145      reverse = 541      d = 3
n = 91       reverse = 19       d = 2
n = 432      reverse = 234      d = 3
n = 43       reverse = 34       d = 2
```

```
#include <stdlib.h>
int reverse(int x, int *digits, int *sum);
int main(){
    int n, r, d;
    FILE *inFile, *outFile;
    inFile = fopen("Numbers.txt", "r");
    if (inFile == NULL){
        printf("Error opening file !");
        exit(1);
    }
    outFile = fopen("Result.txt", "w");
    while(fscanf(inFile, "%d", &n) != EOF){
        if(reverse(n, &d, &r) == 1){
            fprintf(outFile, "n = %5d \treverse = %5d \td = %5d\n", n,r,d);
            fprintf(stdout, "n = %5d \treverse = %5d \td = %5d\n", n,r,d);
        }
    }
    fclose(inFile);
    fclose(outFile);
    return 0;
}

int reverse(int x, int *digits, int *reverse){
    int r;
    *reverse = 0;
    *digits = 0;
    if(x <= 9) // Number negative or has one digit
        return (0);
    while(x != 0){
        *digits +=1; // Don't use *digits++
        r = x % 10;
        *reverse = *reverse*10 + r;
        x = x / 10;
        if(r == 0) // Number has a 0 digit
            return (0);
    }
    return(1);
}
```