Vg101. Introduction to Computer and Programming (Fall 2018)

Homework #5

Assigned: 10/30/2018 Due: 11/6/2018

Notes for submission: We are going to use the online judge (OJ) to grade your homework, so please submit your homework in the OJ. Please see the instruction for OJ submission on Canvas Announcement. For backup purpose, you should also submit your homework to the Canvas. **For canvas submission, please name the file using the format bellow: "(Your last name)(Initial of your first name)_sYourStudentID_midterm1.zip" which includes all your source codes. For example, WuJ_s12345678_hw5.zip**. And the source file name for each problem should be Problem1.c, Problem2.c, etc.

All of the following programs should be written in C

1. Read the following program and complete the function "SecondSmallest" (highlighted in red) to find the second smallest number in an integer array. Here we assume that the number of elements in the array is >= 2 and all the numbers are different.

```
#include <stdio.h>
#include <string.h>

#define LENGTH 1000 /* max input characters */
#define N 100 /* max number of elements in the array */

/* finish this function */
int SecondSmallest(int array[], int n)
{ /* find the second smallest number in the array, where n is the number of elements */
}

int main()
{
        char str[LENGTH]; /* max input characters */
        int arr[N]; /* less than N numbers */
        int i, n, start, end, len;

        for (i=0; i<LENGTH; i++) str[i] = 0; /* initialize */
        printf("Input numbers: ");
        scanf("%[^\n]", str);
        len = strlen(str); /* length of str */
        start = n = 0; /* initialize */
        while (start < len)
        {
```

```
            /* skip spaces */
            while (start < len && str[start] == ' ') start++;
            if (start == len) break; /* end of string */
            /* find end of the number */
            end = start;
            while (end < len && str[end] != ' ') end++;
            /* convert str[start..end] to number */
            arr[n] = 0;
            for (i = start; i < end; i++) arr[n] = arr[n] * 10 + str[i] - '0';
            n++; /* number of elements */
            start = end + 1;
        }
        /* call the function SecondSmallest */
        printf("The second smallest number is %d\n", SecondSmallest(arr, n));
        return 0;
}
```

An example run of this program looks like:

```
jigang@JI:~$ ./a.out
Input numbers: 1 432 5 32 432
The second smallest number is 5
```

2. We know that integers can be represented using different integer types, such as “int”, “long”, and so on. But generally we cannot use a simple type to represent integers that larger than 2^{64}-1, which is more than 64-bit. One solution in this case is to use an array, with each element corresponds to one digit of the integer. Write a program to ask the user to type in two integers, which can be longer than 64-bit, then calculate and show their sum. For simplicity, let’s consider only positive integers, and the number of digit is less than 255. The following is an example run of the program:

```
1st integer? 12345678912345678912345678
2nd integer? 12345678912345678912345678
Their sum is 24691357824691357824691357
```