Lab 2 – Functions and Pointers

Note: You are required to submit your lab code as part of assignment submission for grading via APAS.

1. **(numDigits)** Write a function that counts the number of digits for a non-negative integer. For example, 1234 has 4 digits. The function **numDigits1()** returns the result. The function prototype is given below:

```
int numDigits1(int num);
```

Write another function **numDigits2()** that passes the result through the pointer parameter, *result*. The function prototype is given below:

```
void numDigits2(int num, int *result);
```

The following sample program template is given for testing the functions:

```
#include <stdio.h>
int numDigits1(int num);
void numDigits2(int num, int *result);
int main()
{
  int number, result=0;
  printf("Enter the number: \n");
  scanf("%d", &number);
  printf("numDigits1(): %d\n", numDigits1(number));
  numDigits2(number, &result);
  printf("numDigits2(): %d\n", result);
  return 0;
}
int numDigits1(int num)
{
    int count = 0;
   do {
     count++;
     num = num/10;
   } while (num > 0);
   return count;
}
void numDigits2(int num, int *result)
{
  /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter the number:
1
numDigits1(): 1
```

numDigits2(): 1

(2) Test Case 2:
Enter the number:
13579
numDigits1(): 5
numDigits2(): 5

2. **(digitPos)** Write the function **digitPos1()** that returns the position of the first appearance of a specified digit in a positive number. The position of the digit is counted from the right and starts from 1. If the required digit is not in the number, the function should return 0. For example, digitPos1(12315, 1) returns 2 and digitPos1(12, 3) returns 0. The function prototype is given below:

```
int digitPos1(int num, int digit);
```

Write another function **digitPos2()** that passes the result through the pointer parameter, *result*. For example, if num = 12315 and digit = 1, then *result = 2 and if num=12 and digit = 3, then *result = 0. The function prototype is given below:

```
void digitPos2(int num, int digit, int *result);
```

For separate program testing: The following sample program template is given for testing the functions:

```
#include <stdio.h>
int digitPos1(int num, int digit);
void digitPos2(int num, int digit, int *result);
int main()
{
  int number, digit, result=0;
  printf("Enter the number: \n");
  scanf("%d", &number);
  printf("Enter the digit: \n");
  scanf("%d", &digit);
  printf("digitPos1(): %d\n", digitPos1(number, digit));
  digitPos2(number, digit, &result);
  printf("digitPos2(): %d\n", result);
  return 0;
}
int digitPos1(int num, int digit)
{
  /* Write your code here */
}
void digitPos2(int num, int digit, int *result)
{
   int pos=0;
   *result=0;
   do {
     pos++;
     if (num%10 == digit){
```

```
            *result = pos;
            break;
          }
          num = num/10;
        } while (num > 0);

    }
```

Some sample input and output sessions are given below:

(1) Test Case 1:
```
Enter the number:
234567
Enter the digit:
6
digitPos1(): 2
digitPos2(): 2
```

(2) Test Case 2:
```
Enter the number:
234567
Enter the digit:
8
digitPos1(): 0
digitPos2(): 0
```

3. **(square)** Write a function **square1()** that returns the square of a positive integer number *num*, by computing the sum of odd integers starting with 1 as shown in the example below. The result is returned to the calling function. For example, if *num* = 4, then $4^2 = 1 + 3 + 5 + 7 = 16$ is returned; if *num* = 5, then $5^2 = 1 + 3 + 5 + 7 + 9 = 25$ is returned. The function prototype is:

```
int square1(int num);
```

Write another function **square2()** that passes the result through the pointer parameter, *result*. For example, if *num* = 4, then *result = $4^2 = 1 + 3 + 5 + 7 = 16$; if *num* = 5, then *result = $5^2 = 1 + 3 + 5 + 7 + 9 = 25$. The function prototype is:

```
void square2(int num, int *result);
```

For separate program testing: The following sample program template is given for testing the functions:

```
#include <stdio.h>
int square1(int num);
void square2(int num, int *result);
int main()
{
  int number, result=0;
  printf("Enter the number: \n");
  scanf("%d", &number);
```

```
    printf("square1(): %d\n", square1(number));
    square2(number, &result);
    printf("square2(): %d\n", result);
    return 0;
}
int square1(int num)
{
    /* Write your code here */
}
void square2(int num, int *result)
{
    /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:
Enter the number:
4
square1(): 16
square2(): 16

(2) Test Case 2:
Enter the number:
0
square1(): 0
square2(): 0