Week 3 Lab Tutorial: Functions and Pointers – Suggested Solutions

Lab Questions

Q1:

(i)

3478	100	p	p = 100
7700	8	number	number = 8

That is (a) number is 8 (b) &number is 7700 (c) p is 100 (d) &p is 3478 (e) *p is the content of the memory location 100.

(ii)

3478	100	p	
7700	100	number	number = p

That is (a) number is 100 (b) &number is 7700 (c) p is 100 (d) &p is 3478 (e) *p is the content of the memory location 100.

(iii)

3478	7700	p	p = &number
7700	100	number	

That is (a) number is 100 (b) &number is 7700 (c) p is 7700 (d) &p is 3478 (e) *p is 100.

(iv)

3478	7700	p	*p = 10
7700	10	number	

That is (a) number is 10 (b) &number is 7700 (c) p is 7700 (d) &p is 3478 (e) *p is 10.

(v)

3478	7700	p	
7700	3478	number	number = &p

That is (a) number is 3478 (b) &number is 7700 (c) p is 7700 (d) &p is 3478 (e) *p is 3478.

(vi)

3478	3478	p	p = &p
7700	3478	number	

That is (a) number is 3478 (b) &number is 7700 (c) p is 3478 (d) &p is 3478 (e) *p is 3478.

lab2.c

```c
#include <stdio.h>

/* function prototypes */
int numDigits1(int num);
int digitPos1(int num, int digit);
int square1(int num);
void numDigits2(int num, int *result);
void digitPos2(int num, int digit, int *result);
void square2(int num, int *result);

int main()
{
   int choice;
   int number, digit, result=0;
   do {
      printf("\nPerform the following functions ITERATIVELY:\n");
      printf("1:  numDigits1()\n");
      printf("2:  numDigits2()\n");
      printf("3:  digitPos1()\n");
      printf("4:  digitPos2()\n");
      printf("5:  square1()\n");
      printf("6:  square2()\n");
      printf("7:  quit\n");
      printf("Enter your choice: ");
      scanf("%d", &choice);

      switch (choice) {
         case 1:
            printf("Enter the number: \n");
            scanf("%d", &number);
            printf("numDigits1(): %d\n", numDigits1(number));
            break;
         case 2:
            printf("Enter the number: \n");
            scanf("%d", &number);
            numDigits2(number, &result);
            printf("numDigits2(): %d\n", result);
            break;
         case 3:
            printf("Enter the number: \n");
            scanf("%d", &number);
            printf("Enter the digit: \n");
            scanf("%d", &digit);
            printf("digitPos1(): %d\n", digitPos1(number, digit));
            break;
         case 4:
            printf("Enter the number: \n");
            scanf("%d", &number);
            printf("Enter the digit: \n");
            scanf("%d", &digit);
            digitPos2(number, digit, &result);
            printf("digitPos2(): %d\n", result);
            break;
         case 5:
            printf("Enter the number: \n");
            scanf("%d", &number);
            printf("square1(): %d\n", square1(number));
            break;
         case 6:
            printf("Enter the number: \n");
            scanf("%d", &number);
            square2(number, &result);
            printf("square2(): %d\n", result);
            break;
         default: printf("Program terminating .....\n");
```

```
            break;
      }
   } while (choice < 7);
   return 0;
}
// Question 2
int numDigits1(int num)
{
   int count = 0;
   do {
      count++;
      num = num/10;
   } while (num > 0);

   return count;
}
void numDigits2(int num, int *result)
{
   *result=0;
   do {
      (*result)++;
      num = num/10;
   } while (num > 0);
}

// Question 3
int digitPos1(int num, int digit)
{
   int pos=0;
   do {
      pos++;
      if (num % 10 == digit)
         return pos;
      num = num / 10;
   } while (num > 0);
   return 0;

}
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);
}

// Question 4
int square1(int num)
{
   int count=0, k=1, result=0;
   while (count < num)
   {
      result += k;
      k += 2;
      count++;
   }
   return result;

}
void square2(int num, int *result)
{
   int count=0, k=1;
```

```
   *result=0;
   while (count < num)
   {
      *result += k;
      k += 2;
      count++;
   }
}
```

Q5:

The output:

	remark
h = 5, k = 15	line (i)
h = -100, k = -100	line (v)
h = 5, k = 15	line (ii)
h = 5, k = 15	line (vi)
h = 100, k = 100	line (vii)
h = 5, k = 15	line (iii)
h = 5, k = 15	line (viii)
h = 200, k = 200	line (ix)
h = 200, k = 200	line (iv)