

Week 3 Lab Tutorial: Functions and Pointers – Suggested Solutions

Lab Questions

Q1 – Q3:

```
#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 1
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 2
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 3
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++;
    }
}

```

Q4:

```

#include <stdio.h>
#include <math.h>
void inputXY(double *x1, double *y1, double *x2, double *y2);
void outputResult(double dist);
double calDistance1(double x1, double y1, double x2, double y2);
void calDistance2(double x1, double y1, double x2, double y2, double *dist);
int main()
{
    double x1, y1, x2, y2, distance;

    inputXY(&x1, &y1, &x2, &y2);           // call by reference
    distance = calDistance1(x1, y1, x2, y2); // call by value
    printf("calDistance1(): ");
    outputResult(distance);
    calDistance2(x1, y1, x2, y2, &distance); // call by reference
    printf("calDistance2(): ");
    outputResult(distance);                // call by value
    return 0;
}
void inputXY(double *x1, double *y1, double *x2, double *y2)
{
    printf("Input x1 y1 x2 y2: \n");
    scanf("%lf %lf %lf %lf", x1, y1, x2, y2);
}
void outputResult(double dist)
{
    printf("%.2f\n", dist);
}
double calDistance1(double x1, double y1, double x2, double y2)
{
    x1 = x1 - x2;
    x1 = x1 * x1;
    y1 = y1 - y2;
    y1 = y1 * y1;
    return (sqrt(x1 + y1));
}
void calDistance2(double x1, double y1, double x2, double y2, double *dist)
{
    x1 = x1 - x2;
    x1 = x1 * x1;
    y1 = y1 - y2;
    y1 = y1 * y1;
    *dist = sqrt(x1 + y1);
}

```