Week 7 Lab Tutorial: Recursive Functions – Suggested Solutions

Lab Questions

```c
#include <stdio.h>
/* function prototypes */
int rNumDigits1(int num);
void rNumDigits2(int num, int *result);
int rDigitPos1(int num, int digit);
void rDigitPos2(int num, int digit, int *pos);
int rSquare1(int num);
void rSquare2(int num, int *result);

int main()
{
   int choice;
   int number;
   int digit, result=0;

   do {
      printf("\nPerform the following functions ITERATIVELY:\n");
      printf("1:  rNumDigits1()\n");
      printf("2:  rNumDigits2()\n");
      printf("3:  rDigitPos1()\n");
      printf("4:  rDigitPos2()\n");
      printf("5:  rSquare1()\n");
      printf("6:  rSquare2()\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("rNumDigits1(): %d\n", rNumDigits1(number));
            break;
         case 2:
            printf("Enter the number: \n");
            scanf("%d", &number);
            rNumDigits2(number, &result);
            printf("rNumDigits2(): %d\n", result);
            break;
         case 3:
           printf("Enter the number: \n");
            scanf("%d", &number);
            printf("Enter the digit: \n");
           scanf("%d", &digit);
            printf("rDigitPos1(): %d\n", rDigitPos1(number, digit));
            break;
         case 4:
            printf("Enter the number: \n");
            scanf("%d", &number);
            printf("Enter the digit: \n");
            scanf("%d", &digit);
            rDigitPos2(number, digit, &result);
            printf("rDigitPos2(): %d\n", result);
            break;
         case 5:
            printf("Enter the number: \n");
```

```
            scanf("%d", &number);
            printf("rSquare1(): %d\n", rSquare1(number));
            break;
         case 6:
            printf("Enter the number: \n");
            scanf("%d", &number);
            rSquare2(number, &result);
            printf("rSquare2(): %d\n", result);
            break;
         default: printf("Program terminating .....\n");
            break;
      }
   } while (choice < 7);
   return 0;
}
/* Question Q1 */
int rNumDigits1(int num)
{
   if (num < 10)
      return 1;
   else
      return rNumDigits1(num/10) + 1;
}
void rNumDigits2(int num, int *result)
{
   if (num < 10)
      *result = 1;
   else {
     rNumDigits2(num/10, result);
      *result = *result + 1;
   }
}
/* Question Q2 */
int rDigitPos1(int num, int digit)
{
   int p;

   if (num % 10 == digit)
      return 1;
   else if (num < 10)
      return 0;
   else {
      p = rDigitPos1(num/10, digit);
      if (p > 0)
         return p + 1;
      else
         return 0;
   }
}
void rDigitPos2(int num, int digit, int *pos)
{
   if (num % 10 == digit)
      *pos = 1;
   else if (num < 10)
      *pos = 0;
   else {
      rDigitPos2(num/10, digit, pos);
     if (*pos > 0)
         *pos += 1;
      else
         *pos = 0;
```

```
   }
}
/* Question Q3 */
int rSquare1(int num)
{
   int result=1;

   if (num == 1)
      return result;
   else
      return rSquare1(num-1) + (2*num -1);
}
void rSquare2(int num, int *result)
{
   if (num == 1)
      *result = 1;
   else {
      rSquare2(num-1, result);
      *result += (2*num -1);
   }
}
```

Q4

```
#include <stdio.h>
#define SIZE 20
int rCountArray(int array[], int n, int a);
int main()
{
   int array[SIZE];
   int index, count, target, size;

   printf("Enter array size: \n");
   scanf("%d", &size);
   printf("Enter %d numbers: \n", size);
   for (index = 0; index < size; index++)
      scanf("%d", &array[index]);
   printf("Enter the target number: \n");
   scanf("%d", &target);
   count = rCountArray(array, size, target);
   printf("rCountArray(): %d\n", count);
   return 0;
}
int rCountArray(int array[], int n, int a)
{
   if  (n == 1)
   {
      if (array[0] == a)
         return 1;
      else
         return 0;
   }
   if (array[0] == a)
      return 1 + rCountArray(&array[1], n-1, a);
   else
      return rCountArray(&array[1], n-1, a);
}
// another version
/*
int rCountArray(int array[], int n, int a)
{
   if  (n == 1)
```

```
   {
      if (array[0] == a)
         return 1;
      else
         return 0;
   }
   if (array[n-1] == a)
      return 1 + rCountArray(&array[0], n-1, a);
   else
      return rCountArray(&array[0], n-1, a);
}
*/
// another version
/*
int rCountArray(int array[], int n, int a)
{
   int count;

   if(n == 0)
      return 0;

   count = rCountArray(array + 1, n - 1, a);
   if(*array == a)
      return count + 1;
   else
      return count;
}
*/
```