Week 7 Tutorial: Recursion – Suggested Solutions

Q1: (rSumup)

```
#include <stdio.h>
int rSumup1(int n);
void rSumup2(int n, int *result);
int main()
{
   int n, result;

   printf("Enter a number: \n");
   scanf("%d", &n);
   printf("rSumup1(): %d\n", rSumup1(n));
   rSumup2(n, &result);
   printf("rSumup2(): %d\n",result);
   return 0;
}
int rSumup1(int n)
{
   if (n == 1)
      return 1;
   else
      return n +  rSumup1(n-1);
}
void rSumup2(int n, int *result)
{
   if (n == 1)
      *result=1;
   else
   {
      rSumup2(n-1, result);
      *result += n;
   }
}
```

Q2: (rDigitValue)

```
#include <stdio.h>
int rDigitValue1(int num, int k);
void rDigitValue2(int num, int k, int *result);
int main()
{
   int k;
   int number, digit;

   printf("Enter a number: \n");
   scanf("%d", &number);
   printf("Enter k position: \n");
   scanf("%d", &k);
   printf("rDigitValue1(): %d\n", rDigitValue1(number, k));
   rDigitValue2(number, k, &digit);
   printf("rDigitValue2(): %d\n", digit);
   return 0;
}
int rDigitValue1(int num, int k)
{
   if (k==0)
      return 0;
   else if (k==1)
      return num%10;
```

```c
   else
      return rDigitValue1(num/10, k -1);
}
void rDigitValue2(int num, int k, int *result)
{
   if (k==0)
      *result = 0;
   else if (k==1)
      *result = num%10;
   else
      rDigitValue2(num/10, k-1, result);
}
```

Q3:

If we execute the program and input to the program “ward ” then the output will be “draw”:

```
Program Output
Enter your word and end it with a space => ward

draw

$
```

Please note that there is a blank character at the end of the input word before the “enter” key is pressed. Basically, this program prints an input string, which ends with a space character, in the reversed order. (tutors: please explain why)

Q4: (rCountArray)

```c
#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;
}
*/
```