Week 5 Tutorial: Character Strings – Suggested Solutions

Q1: (processString)

```
#include <stdio.h>
#include <string.h>
void processString(char *str, int *totVowels, int *totDigits);
int main()
{
   char str[50], *p;
   int totVowels, totDigits;

   printf("Enter the string: \n");
   fgets(str, 50, stdin);
   if (p=strchr(str,'\n')) *p = '\0';
   processString(str, &totVowels, &totDigits);
   printf("Total vowels = %d\n", totVowels);
   printf("Total digits = %d\n", totDigits);
   return 0;
}
void processString(char *str, int *totVowels, int *totDigits)
{
   int i, size;

   *totVowels=0;
   *totDigits=0;
   i=0; size=0;
   while (str[i]!='\0'){
      size++;
      i++;
   }
   for (i=0; i < size; i++) {
      if (str[i] == 'a' || str[i] == 'e' ||
            str[i] == 'i' ||  str[i] == 'o' ||
             str[i] == 'u' || str[i] == 'A' ||
             str[i] == 'E' || str[i] == 'I' ||
             str[i] == 'O' || str[i] == 'U')
         (*totVowels)++;
      else if ( str[i] >= '0' &&  str[i] <= '9')
         (*totDigits)++;
   }
}
/*  another version */
/*
void processString2(char *str, int *totVowels, int *totDigits)
{
   int i,size;

   *totVowels = 0, *totDigits = 0;
   i=0; size=0;
   while (str[i]!='\0'){
      size++;
      i++;
   }
   for (i=0; i < size; i++) {
      if (*(str+i) == 'a' || *(str+i) == 'e' ||
            *(str+i) == 'i' || *(str+i) == 'o' ||
            *(str+i) == 'u' || *(str+i) == 'A' ||
            *(str+i) == 'E' || *(str+i) == 'I' ||
            *(str+i) == 'O' || *(str+i) == 'U')
         (*totVowels)++;
      else if ( *(str+i) >= '0' &&  *(str+i) <= '9')
         (*totDigits)++;
  }
}
*/
```

Q2: (stringncpy)

```
#include <stdio.h>
#include <string.h>
char *stringncpy(char *s1, char *s2, int n);
int main()
{
   char targetStr[40], sourceStr[40], *target, *p;
   int length;

   printf("Enter the string: \n");
   fgets(sourceStr, 40, stdin);
   if (p=strchr(sourceStr,'\n')) *p = '\0';
   printf("Enter the number of characters: \n");
   scanf("%d", &length);
   target = stringncpy(targetStr, sourceStr, length);
   printf("stringncpy(): %s\n", target);
   return 0;
}

char *stringncpy(char *s1, char *s2, int n)
{
   int k, h;

   for (k = 0; k < n; k++)   {
      if (s2[k] != '\0')
         s1[k] = s2[k];
      else
         break;
   }
   s1[k] = '\0';
   for (h = k; h < n; h++)
      s1[h] = '\0';
   return s1;
}
/* another version - using pointer */
/*
char *stringncpy(char *s1, char *s2, int n)
{
   int k, h;

   for (k = 0; k < n; k++)   {
      if (*(s2+k) != '\0')
         *(s1+k) = *(s2+k);
      else
         break;
   }
   *(s1+k) = '\0';
   for (h = k; h < n; h++)
      *(s1+h) = '\0';
   return s1;
}
*/
```

Q3: (stringcmp)

```
#include <stdio.h>
#include <string.h>
#define INIT_VALUE 999
int stringcmp(char *s1, char *s2);
int main()
{
   char source[80], target[80], *p;
   int result = INIT_VALUE;

   printf("Enter a source string: \n");
   fgets(source, 80, stdin);
   if (p=strchr(source,'\n')) *p = '\0';
   printf("Enter a target string: \n");
   fgets(target, 80, stdin);
   if (p=strchr(target,'\n')) *p = '\0';
   result = stringcmp(source, target);
   if (result == 1)
      printf("stringcmp(): greater than");
   else if (result == 0)
      printf("stringcmp(): equal");
   else if (result == -1)
      printf("stringcmp(): less than");
   else
      printf("stringcmp(): error");
   return 0;
}
int stringcmp(char *s1, char *s2)
{
   while (1) {
      if (*s1 == '\0' && *s2 == '\0')
         return 0;
      else if (*s1 == '\0')
         return -1;
      else if (*s2 == '\0')
         return 1;
      else if (*s1 < *s2)
         return -1;
      else if (*s1 > *s2)
         return 1;
      s1++;
      s2++;
   }
}
```

Q4:

The program prints:

```
How are ya, sweetie?Beat the clock.
eat the clock.
win a toy.
win a snoopy.
win
chat
hat
at
t
t
at
How are ya, sweetie?
```