SOLUTION TO ASSIGNMENT 10
C-STRING CODING PROBLEM

The following source code provides *one* solution for the programming Assignment 10.

```
/*************************************************
* Program: a10.cpp - Project 10 - Coding Solution   *
* Written by: R. Gibson   -   Date: 4/24/2012       *
*************************************************/

/* PREPROCESSOR DIRECTIVES */
#include <iostream>  // Header to allow use of console functions
#include <cstring>   // Header to allow use of character functions
using namespace std;

int main ()
{

/* PREPROCESSOR DIRECTIVES */

#define NSIZE 15    // Maximum size of input names: FIRST and LAST
#define SSNSIZE 9   // Maximum size of Soc. Sec. Number
#define USIZE 8     // Maximum size of Username
#define PSIZE 9     // Maximum size of Password

/* LOCAL VARIABLE DECLARATION */

char FIRST [NSIZE+1];    // User's First Name
char LAST [NSIZE+1];     // User's Last Name
char SSN [SSNSIZE+1];    // User's Soc. Sec. Number
char UNAME [USIZE+1];    // User's Account Name
char PWORD [PSIZE+1];    // User's Password

int  PS;  // Index of character position in source strings
int  PT;  // Index of character position in result strings

/* PROCESS DEFINITION */

/* Display program title and credits as per Sample Softcopy */
cout << "Account Generating Program.\n";
cout << "Written by Sam Student - 11/1/2011\n\n";

/* Request and store user input from the keyboard */
cout << "First Name? ";
// Accept up to 15 characters w/whitespace
   cin.getline(FIRST,NSIZE+1); for (PS=0; PS<strlen(FIRST); PS++)
   if (isupper(FIRST[PS])) FIRST[PS]=tolower(FIRST[PS]);
```

```
cout << "Last Name? ";
// Accept up to 15 characters w/whitespace
   cin.getline(LAST,NSIZE+1); for (PS=0; PS<strlen(LAST); PS++)
   if (isupper(LAST[PS])) LAST[PS]=tolower(LAST[PS]);

do
  /* This loop structure was not explicitly specified in the
     algorithm, but the need for it could be inferred from
     the mention of "valid" input data in step III */
  {
   cout << "Social Security # (digits only)? ";
   cin.getline (SSN,SSNSIZE+1);  // store up to 9 numerals only
  }
while ( strlen(SSN) != SSNSIZE );

/* Build User Name from last name and SSN */
   PT=0; // Initialize index for UNAME
/* Copy the first 4 char's of LAST to UNAME */
   for (PS=0; PS<4 && LAST[PS]!='\0'; PS++)
       {UNAME[PT]=LAST[PS]; PT++;}
/* Append last 4 char's of SSN to UNAME using PT from above */
   for (PS=5; PS<SSNSIZE; PS++)
       {UNAME[PT]=SSN[PS]; PT++;}
UNAME[PT]='\0'; // Store an end-of-string after Username

/* Build Password from SSN and first name */
   PT=0; /* Initialize index for PWORD */
/* Copy the first 5 char's of SSN to PWORD */
   for (PS=0; PS<5; PS++) {PWORD[PT]=SSN[PS]; PT++;}
/* Append first 4 char's of FIRST to PWORD using PT from above */
   for (PS=0; PS<4 && FIRST[PS]!='\0'; PS++)
       {PWORD[PT]=FIRST[PS]; PT++;}
PWORD[PT]='\0'; // Store an end-of-string after Password

/* Display and identify Username and Password */
cout << "\nUsername: " << UNAME << endl;
cout << "Password: " << PWORD << endl;

return 0;  // Send a null error code to the parent process
}
```