CS 210 **October 27, 2016**
Day 16

Go over palindrome problem

A palindrome is a word or a sentence which reads the same forward or backward. For example, *mom* is a palindrome as are all of the following (from Wikipedia):
redivider, noon, civic, radar, level, rotor, kayak, reviver, racecar, redder, madam, and refer.

Write a program to input a sentence on one line from the user and determine if the sentence is a palindrome. Your program should print whether or not the input sentence is a palindrome.

Turn in a hard copy of your source code.

SOLUTION 1

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
/* This version compares characters in a sentence,
    first with last, next with next to last, etc. to
    determine if the sentence is palindomic.
*/
int main()
{
    char line[80];
    int i, len;
    int IsPalindrome = 1;
    printf("Input a one line sentence... \n");
    gets(line);
    len = strlen(line);
    for(i=0;i<len;i++)
       line[i] = tolower(line[i]);
    for(i=0;i<len/2;i++)
    {
        if(line[i] != line[len-i-1])
           IsPalindrome = 0;
    }
    if(IsPalindrome)
        printf("%s is a palindrome.\n", line);
    else
        printf("%s is NOT a palindrome.\n", line);
}
```

SOLUTION 2

```
/* This version creates a new reversed sentence and
    compares it to the original to determine if the
    sentence is palindromic.
*/
int main()
{
    char line[80];
    char pal[80];
    int i, len;
    int IsPalindrome = 1;
    printf("Input a one line sentence... \n");
    gets(line);
    len = strlen(line);
    for(i=0;i<len;i++)
       line[i] = tolower(line[i]);
    for(i=0;i<len;i++)
    {
         pal[i] = line[len-i-1];
    }
    pal[i] = '\0';
    if(strcmp(line, pal) == 0)
        printf("%s is a palindrome.\n", line);
    else
        printf("%s is NOT a palindrome.\n", line);
}
```

STRTOK – String token

The strtok string function in string.h breaks a string into tokens where tokens are arbitrary characters defined by the user. For example, if I have a string "Mud and chocolate don't mix.", I could define a token a space " ". strtok could then be used to break this string into five words separated by spaces.

Here is an example: Track through this example with the debugger.

```
#include<stdio.h>
#include<string.h>

int main ()
{
  char str[] ="- This, a sample string.";
  char *pch;
  char *context;
  const char sTokens[] = " ,.-";
  pch = strtok_s(str, sTokens, &context);
  while (pch != NULL)
  {
    printf ("%s\n",pch);
    pch = strtok_s(NULL, sTokens, &context);
  }
  return 0;
}
```

Or, if you want to use the unsafe strtok:

```
#include<stdio.h>
#include<string.h>
#pragma warning(disable:4996)

int main ()
{
  char str[] ="- This, a sample string.";
  char *pch;
  char sTokens[] = " ,.-";
  pch = strtok (str, sTokens);
  while (pch != NULL)
  {
    printf ("%s\n",pch);
    pch = strtok(NULL, sTokens);
  }
  return 0;
}
```

Sample 2-D array declaration.

```
int grid[10][7];  //Declare a two-D grid
```

Sample prototype for a 2D array. Only second dimension is specified

```
void MyFunction(int grid[][10]);
```

Sample function definition for passing grid.

```
void MyFunction(int grid[][10])
{
   Grid[i][j] = value;
}
```

Sample 3-D array declaration.

```
int grid[10][7][3];  //Declare a three-D grid
```

Sample prototype for a 2D array. Only second dimension is specified

```
void MyFunction(int grid[][7][3]);
```

Sample function definition for passing grid.

```
void MyFunction(int grid[][7][3])
{
   Grid[i][j][k] = value;
}
```

Initializing a 1D array

```
int d1[] = {1, 2, 3, 4, 5};
```

Initializing a 2D array

```
int d2[][4] = {{1, 2, 3, 4, 5},
               {6, 7, 8, 9, 0},
               {1, 2, 3, 4, 5},
               {6, 7, 8, 9, 0}
              };
```

Initializing a 3D array, 2 rows, 2 columns, and 5 planes.

```
int d3[][2][5] = {{{1, 2, 3, 4, 5}, {6, 7, 8, 9, 0}},
                  {{0, 9, 8, 7, 6}, {5, 4, 3, 2, 1}}
                 };
```

Example problem – student grades

```
#include<stdio.h>
void PrintAll(int grades[][4]);
void PrintStudentAvg(int grades[][4]);
void PrintExamAvg(int grades[][4]);
int main()
{
 int grades[][4] = {{99, 90, 50, 94},
                    {88, 78, 87, 95},
                    {80, 78, 70, 80},
                    {75, 76, 77, 78},
                    {69, 79, 89, 99}
                   };
 PrintAll(grades);
 PrintStudentAvg(grades);
 PrintExamAvg(grades);
}
void PrintAll(int grades[][4])
{
   int r, c;
   for(r=0;r<5;r++)
   {
       for(c=0;c<4;c++)
          printf("%d, ", grades[r][c]);
       printf("%\n");
   }
}
void PrintStudentAvg(int grades[][4])
      {int r, c, sum;
       double avg;
       for(r=0;r<5;r++)
         {sum = 0;
          for(c=0;c<4;c++)
             sum += grades[r][c];
          avg = (double)sum/4;
          printf("Student %d %6.2f\n", r, avg);
         }
      }
void PrintExamAvg(int grades[][4])
      {int r, c, sum;
       double avg;
       for(c=0;c<4;c++)
         {sum = 0;
          for(r=0;r<5;r++)
             sum += grades[r][c];
          avg = (double)sum/5;
          printf("Test %d %6.2f\n", c, avg);
         }
      }
```

Prints the following:

```
99, 90, 50, 94,
88, 78, 87, 95,
80, 78, 70, 80,
75, 76, 77, 78,
69, 79, 89, 99,
Student 0  83.25
Student 1  87.00
Student 2  77.00
Student 3  76.50
Student 4  84.00
Test 0  82.20
Test 1  80.20
Test 2  74.60
Test 3  89.20
Press any key to continue . . .
```

CS 210 **October 25, 2016**
Multidimensional Arrays

Write a program in the c-language to define a 3D array as shown:

```
int d3[][3][5] = {{{1, -2, 3, -4, 5}, {6, -7, 8, 9, 0}},
                   {{10, -9, 8, 7, 6}, {5, 4, -3, 2, 1}},
                   {{6, -9, 3, -2, 6}, {1, 8, -6, 1, 5}}
                  };
```

Your program should go through the array and change all negative numbers to zero. Print the final array to the console.

Turn in a hard copy of source code.

Sample results:

```
1, 0, 3, 0, 5,
6, 0, 8, 9, 0,
0, 0, 0, 0, 0,

10, 0, 8, 7, 6,
5, 4, 0, 2, 1,
0, 0, 0, 0, 0,

Press any key to continue . . .
```