McMaster University

Principles of Programming Tutorial Exercise

Exercise Sheet- Week8

1- (**Fahrenheit to Celsius)** Write a program that converts integer Fahrenheit temperatures from 0 to 212 degrees to floating-point Celsius temperatures with 3 digits of precision. Use the formula :

$$celsius = 5.0 / 9.0 * (fahrenheit - 32);$$

to perform the calculation. The output should be printed in two right-justified columns of 10 characters each, and the Celsius temperatures should be preceded by a sign for both positive and negative values.

```
#include <stdio.h>

int main()
{
      int fahrenheit; /* holds fahrenheit temperature */
      double celcius; /* holds celcius temperature */

      printf( "%10s%12s\n", "Fahrenheit", "Celcius" );

      /* convert fahrenheit to celsius and display temperatures
       showing the sign for celsius temperatures */
      for ( fahrenheit = 0; fahrenheit <= 212; fahrenheit++ ) {
            celcius = 5.0 / 9.0 * ( fahrenheit - 32 );
            printf( "%10d%+12.3f\n", fahrenheit, celcius );
      } /* end for */
```

Output:

```
Fahrenheit Celcius

0 -17.778
1 -17.222
2 -16.667
3 -16.111
4 -15.556
5 -15.000
6 -14.444
7 -13.889
.
.
.
204 +95.556
205 +96.111
206 +96.667
207 +97.222
208 +97.778
209 +98.333
210 +98.889
211 +99.444
212 +100.000
```

2- (Scape sequences)Write a program to test all the escape sequences in Figure 9.16. For the escape sequences that move the cursor, print a character before and after printing the escape sequence so it is clear where the cursor has moved.

```c
#include <stdio.h>

int main()
{

 /* test all escape sequences */
 printf( "The single quote : \'\n" );
 printf( "The double quote : \"\n" );
  printf( "The question mark: \?\n" );
  printf( "The backslash : \\\n" );

  printf( "The bell. \a\n\n" );

  printf( "Move cursor back one position on current line. *\b*\n" );
  printf( "Move cursor to start of next logical page. *\f*\n" );

  printf( "Move cursor to the beginning of next line. *\n*\n" );
  printf( "Move cursor to the beginning of current line. *\r*\n" );

  printf( "Move cursor to the next horizontal tab position. *\t*\n" );
  printf( "Move cursor to the next vertical tab position. *\v*\n" );

  return 0; /* indicate successful termination */

 } /* end main */
```

Output

```
The single quote : '

The double quote : "
The question mark: ?
The backslash : \
The bell.
Move cursor back one position on current line. *
Move cursor to start of next logical page. *?*
Move cursor to the beginning of next line. *
*
*ove cursor to the beginning of current line. *
Move cursor to the next horizontal tab position. * *
Move cursor to the next vertical tab position. *?*
```

3- (**Union**)Create union integer with members char c, short s, int i and long b. Write a program that inputs value of type char, short, int and long and stores the values in union variables of type union integer. Each union variable should be printed as a char, a short, an int and a long. Do the values always print correctly?

```
/* Exercise 10.8 Solution */
/* NOTE: The program output is machine dependent */
 #include <stdio.h>
/* integer union definition */
 union integer {
 char c; /* character input by user */
 short s; /* short integer input by user */
 int i; /* integer input by user */
  long l; /* long integer input by user */
 }; /* end union integer */

 int main()
 {
  union integer a; /* define union a */
  /* read a character from user into the union */
  printf( "Enter a character: " );
  scanf( "%c", &a.c );

  /* print each value of union */
  printf( "\'%c'\ printed as a character is %c\n", a.c, a.c );
  printf( "\'%c'\ printed as a short integer is %hd\n", a.c, a.s );
  printf( "\'%c'\ printed as an integer is %d\n", a.c, a.i );
  printf( "\'%c'\ printed as a long integer is %ld\n", a.c, a.l );

  /* read a short integer from user into the union */
  printf( "\nEnter a short integer: " );
  scanf( "%hd", &a.s );

  /* print each value of union */
  printf( "%hd printed as a character is %c\n", a.s, a.c );
  printf( "%hd printed as a short integer is %hd\n", a.s, a.s );
  printf( "%hd printed as an integer is %d\n", a.s, a.i );
  printf( "%hd printed as a long integer is %ld\n", a.s, a.l );

  /* read an integer from user into the union */
  printf( "\nEnter an integer: " );
  scanf( "%d", &a.i );

  /* print each value of union */
  printf( "%d printed as a character is %c\n", a.i, a.c );
  printf( "%d printed as a short integer is %hd\n", a.i, a.s );
  printf( "%d printed as an integer is %d\n", a.i, a.i );
  printf( "%d printed as a long integer is %ld\n", a.i, a.l );

  /* read a long integer from user into the union */
  printf( "\nEnter a long integer: " );
  scanf( "%ld", &a.l );

  /* print each value of union */
  printf( "%ld printed as a character is %c\n", a.l, a.c );
  printf( "%ld printed as a short integer is %hd\n", a.l, a.s );
  printf( "%ld printed as an integer is %d\n", a.l, a.i );
  printf( "%ld printed as a long integer is %ld\n", a.l, a.l );
  return 0; /* indicate successful termination */

 } /* end main */
```

Output

```
Enter a character: A

'A' printed as a character is A
'A' printed as a short integer is -13247
'A' printed as an integer is -858993599
'A' printed as a long integer is -858993599
Enter a short integer: 97
97 printed as a character is a
97 printed as a short integer is 97
97 printed as an integer is -859045791
97 printed as a long integer is -859045791
Enter an integer: 32700
32700 printed as a character is +
32700 printed as a short integer is 32700
32700 printed as an integer is 32700
32700 printed as a long integer is 32700
Enter a long integer: 10000000
10000000 printed as a character is Ç
10000000 printed as a short integer is -27008
10000000 printed as an integer is 10000000
10000000 printed as a long integer is 10000000
```

4- (**Character constant**)Write a program that determines whether ? can be printed as the character constant '?' rather than the character constant scape sequence '\?' using conversion specifier %c in the format control string of a printf statement.

```
/* Exercise 9.19 Solution */
#include <stdio.h>

int main()
{
 const char questionMark = '?'; /* define '?' as a char constant */

 printf( "This %c can be printed without using the \\\?\n",
 questionMark );

  return 0; /* indicate successful termination */

 } /* end main */
```

Output:

```
This ? can be printed without using the \?
```