

```

/* Question 1 Solution */
#include <stdio.h>
#include <math.h>

void calculateFloor( void ); /* function prototype */

int main()
{
    calculateFloor(); /* call function calculateFloor */

    return 0; /* indicate successful termination */
} /* end main */

/* calculateFloor rounds 5 inputs */
void calculateFloor( void )
{
    double x; /* current input */
    double y; /* current input rounded */
    int loop; /* loop counter */

    /* loop for 5 inputs */
    for ( loop = 1; loop <= 5; loop++ )
    {
        printf( "Enter a floating-point value: " );
        scanf( "%lf", &x );

        /* y holds rounded input */
        y = floor( x + .5 );
        printf( "%f rounded is %.1f\n\n", x, y );
    } /* end for */
} /* end function calculateFloor */

```

```

/* Question 2 Solution */
#include <stdio.h>
#include <math.h>

double calculateCharges( double hours ); /* function prototype */

int main()
{
    double h; /* number of hours for current car */
    double currentCharge; /* parking charge for current car */
    double totalCharges = 0.0; /* total charges */
    double totalHours = 0.0; /* total number of hours */
    int i; /* loop counter */
    int first = 1; /* flag for printing table headers */

    printf( "Enter the hours parked for 3 cars: " );
    /* loop 3 times for 3 cars */
    for ( i = 1; i <= 3; i++ )
    {
        scanf( "%lf", &h );
        totalHours += h; /* add current hours to total hours */

        /* if first time through loop, display headers */
        if ( first )
        {
            printf( "%5s%15s%15s\n", "Car", "Hours", "Charge" );

            /* set flag to false to prevent from printing again */
            first = 0;
        } /* end if */

        /* calculate current car's charge and update total */
        totalCharges += ( currentCharge = calculateCharges( h ) );

        /* display row data for current car */
        printf( "%5d%15.1f%15.2f\n", i, h, currentCharge );
    } /* end for */

    /* display row data for totals */
    printf( "%5s%15.1f%15.2f\n", "TOTAL", totalHours, totalCharges );
    return 0; /* indicate successful termination */
} /* end main */

```

```

/* calculateCharges returns charge according to number of hours */
double calculateCharges( double hours )
{
    double charge; /* calculated charge */

    /* $2 for up to 3 hours */
    if ( hours < 3.0 )

```

```
{
    charge = 2.0;
}/* end if */

/* $.50 for each hour or part thereof in excess of 3 hours */
else if ( hours < 19.0 )
{
    charge = 2.0 + .5 * ceil( hours - 3.0 );
}/* end else if */
else /* maximum charge $10 */
{
    charge = 10.0;
}/* end else */

return charge; /* return calculated charge */
}/* end function calculateCharges */
```

```

/* Question 3 Solution */
#include <stdio.h>
#include <math.h>

double hypotenuse( double s1, double s2 ); /* function prototype */

int main()
{
    int i; /* loop counter */
    double side1; /* value for first side */
    double side2; /* value for second side */

    /* loop 3 times */
    for ( i = 1; i <= 3; i++ )
    {
        printf( "Enter the sides of the triangle: " );
        scanf( "%lf%lf", &side1, &side2 );

        /* calculate and display hypotenuse value */
        printf( "Hypotenuse: %.1f\n\n", hypotenuse( side1, side2 ) );
    } /* end for */

    return 0; /* indicate successful termination */
} /* end main */

/* hypotenuse calculates value of hypotenuse of a right triangle given two side values */
double hypotenuse( double s1, double s2 )
{
    return sqrt( pow( s1, 2 ) + pow( s2, 2 ) );
} /* end function hypotenuse */

```

```

/* Question 4 Solution */
#include <stdio.h>

int celsius( int fTemp ); /* function prototype */
int fahrenheit( int cTemp ); /* function prototype */

int main()
{
    int i; /* loop counter */

    /* display table of Fahrenheit equivalents of Celsius temperature */
    printf( "Fahrenheit equivalents of Celsius temperatures:\n" );
    printf( "Celsius\t\tFahrenheit\n" );

    /* display Fahrenheit equivalents of Celsius 0 to 100 */
    for ( i = 0; i <= 100; i++ )
    {
        printf( "%d\t\t%d\n", i, fahrenheit( i ) );
    } /* end for */

    /* display table of Celsius equivalents of Fahrenheit temperature */
    printf( "\nCelsius equivalents of Fahrenheit temperatures:\n" );
    printf( "Fahrenheit\tCelsius\n" );

    /* display Celsius equivalents of Fahrenheit 32 to 212 */
    for ( i = 32; i <= 212; i++ )
    {
        printf( "%d\t\t%d\n", i, celsius( i ) );
    } /* end for */
    return 0; /* indicate successful termination */
} /* end main */

/* celsius returns Celsius equivalent of fTemp, given in Fahrenheit */
int celsius( int fTemp )
{
    return ( int ) ( 5.0 / 9.0 * ( fTemp - 32 ) );
} /* end function celsius */

/* fahrenheit returns Fahrenheit equivalent of cTemp, given in Celsius */
int fahrenheit( int cTemp )
{
    return ( int ) ( 9.0 / 5.0 * cTemp + 32 );
} /* end function fahrenheit */

```