

```

// J Hundley
// assign06
// Mar 7, 2012
/*
Read the heights of mountains on Earth and Mars from a text file.
Compute and print the mountain distance from the horizon.
Print the highest mountain on each planet.
*/
#include <stdio.h>
#include <math.h>
#define EARTH_DIAMETER 7926 // miles
#define MARS_DIAMETER 4217 // miles
#define FILENAME "mountainHeights.txt"

// FUNCTION PROTOTYPES=====
double km2miles( double km );
double feet2miles( double feet );
double computeHorizonDist( double diamter, double height );
void getPlanetInfo( int planetNum, double mtnHt, double *height, double *diameter,
double *maxMarsHt, double *maxEarthHt );
void printHeaders();
void printMaxHeights( double maxMarsHt, double maxEarthHt );
// =====
int main()
{
    int    numMtn,          // number of planets (input)
           n,              // loop control
           planetNum;      // ID number for planet (input)
    double mtnHt,          // height of mountain in km or feet (input)
           height,         // height of mountain in miles
           diameter,       // radius of planet in miles
           horizonDist;    // distance from mountain to horizon in miles (output)
    // initialize counter and accumulators
    double maxMarsHt = 0, // max mountain height on Mars in miles (output)
           maxEarthHt = 0; // max maountain height on Earth in miles (output)

    FILE * filePtr;

    // prepare file to read
    filePtr = fopen(FILENAME, "r");
    if (filePtr == NULL) printf("FILE OPEN ERROR. END PROGRAM.\n");
    else // good file open
    {
        // table title and column headers
        printHeaders();

        // read first value in file; number of mountains
        fscanf(filePtr, "%d", &numMtn);

        // read the values from data file using fscanf.
        for ( n=0; n<numMtn; n++ )
        {
            fscanf(filePtr, "%d", &planetNum);
            fscanf(filePtr, "%lf", &mtnHt);

            getPlanetInfo(planetNum, mtnHt, &height, &diameter, &maxMarsHt, &maxEarthHt);

            // compute distance from horizon
            horizonDist = computeHorizonDist( diameter, height );

            // print table with planet number and
            // mountain height and distance from horizon in miles
            printf( " %d          %6.1f          %6.1f \n", planetNum, height, horizonDist );
        } // for loop
    }
}

```

```

    // table footer
    printf( "* Units = miles\n\n" );

    // print highest mountain height for each planet
    printMaxHeights( maxMarsHt, maxEarthHt );

} //end else

return 0;
}
// FUNCTION DEFINITIONS=====
// convert kilometers to miles
double km2miles( double km )
{
    return km / 1.609;
}

// convert feet to miles
double feet2miles( double feet )
{
    double miles;           // local variable
    miles = feet / 5280;
    return miles;
}

// compute the distant to the horizon
double computeHorizonDist( double diameter, double height )
{
    return sqrt(diameter * height + pow(height,2));
}

// get information for planet
void getPlanetInfo( int planetNum, double mtnHt, double *height, double *diameter,
                   double *maxMarsHt, double *maxEarthHt )
{
    switch (planetNum)
    {
        case 4: // Mars, mtnHt in km
            *height = km2miles( mtnHt );
            if ( *height > *maxMarsHt )
                *maxMarsHt = *height;
            *diameter = MARS_DIAMETER;
            // printf("mtnHt=%.2f ht=%.2f\n",mtnHt,*height);

            break;
        case 3: // Earth, mtnHt in feet
            *height = feet2miles( mtnHt );
            if ( *height > *maxEarthHt )
                *maxEarthHt = *height;
            *diameter = EARTH_DIAMETER;
            break;
    } // end switch
}

// print title and column headers
void printHeaders()
{
    printf("Distance from Mountain Peak to Horizon\n");
    printf("Planet    Mountain    Horizon          \n");
    printf("Number    Height*    Distance*          \n");
    printf("-----    -\n");
}

// print highest mountain height for each planet
void printMaxHeights( double maxMarsHt, double maxEarthHt )
{
    printf( "The highest mountain on Mars: %.1f miles\n",maxMarsHt );
    printf( "The highest mountain on Earth: %.1f miles\n",maxEarthHt );
}

```

COMP1200-C - assign 06  
Due midnight – Wednesday – March 21  
**Submit** assign06.c and mountainHeights.txt **via Blackboard**

**Before you start writing your program:**

Your devPlan05 still applies to your solution to the assign06 assignment problem.  
Save your assign05.c as assign06.c and edit as required.

*NOTE: Your submitted file(s) MUST be spelled and cased as instructed.*

**Program: assign06.c**

You are to modify your assign05.c to use user-defined functions. You are to use the following prototypes. The return type, function name, formal parameters, and data type should stay the same. You may use your own variable names, but the same information and must be represented in the list.

*Read all instructions before beginning your work.*

With the exception of the conversion functions, each function is called (or used) by main. The conversion functions are called by getPlanetInfo.

The prototypes should be placed before your main function, and the functions definitions should be after the main function.

```
// FUNCTION PROTOTYPES=====
// convert kilometers to miles
double km2miles( double km );
// convert feet to miles
double feet2miles( double feet );
// compute the distant to the horizon
double computeHorizonDist( double diamter, double height );
    (The first three function have "Call-By-Value" formal parameters. Only one value is sent back to the calling
    function using the return statement.)
// get information for planet
void getPlanetInfo( int planetNum, double mtnHt, double *height, double *diameter,
    double *maxMarsHt, double *maxEarthHt );
    (This function should contain the selection that checked for the planet number and determined the height in miles
    and diameter. The appropriate maximum height should be updated, also. The "Call-By-Value" formal parameters
    receive a copy of the values sent from the calling function. The "Call-By-Reference" pointer variables will save the
    values at the address where they point. Keep in mind that a pointer variable name receives an address. This function
    will call (or use) the conversion functions.)
// print title and column headers
void printHeaders();
    (This function only prints the information in it; there are no input and output via parameters.)
// print highest mountain height for each planet
void printMaxHeights( double maxMarsHt, double maxEarthHt );
    (This function receives information but does not send a value back to the calling function.)
```

**Problem Constants:**

**Problem Inputs:**

**Problem Outputs:**

**Other variables:**

**Equations:**

**Algorithm:**

(Use the algorithm in your development plan as comments in your program.)

*New commands*  
modularity  
user-defined function  
prototype  
call-by-value parameter  
call-by-reference parameter  
pointer type variable  
return type

### Sample Input/Output:

Same as assign05

### General Instructions:

- ☐ Insert comments at the top and throughout each file
  - Include the follow comments at the beginning of this (and ALL) files.
    - // your name
    - // assignment number
    - // date you completed the assignment
    - // **statement(s) about collaboration**
    - // a short narrative about what the file does
  - Use the algorithm as comments throughout each file
- ☐ Use descriptive variable names.
- ☐ Use Sample Input/Output as a guide.
  - Save the name of the data file as a CONSTANT.
  - Use title and column headers
  - One decimal place for height and distance
  - Print column numbers right-aligned
- ☐ Divide you solution program code into sections as noted in the algorithm.  
Use section comments as well as the algorithm step comments.
- ☐ Indent all blocks.

*-7 points per file for absence of any of these required comments at the top*

### Submit via Blackboard:

assign06.c  
mountainHeights.txt

program file  
The data file that you downloaded  
needs to be submitted so that there  
is a copy in your submission folder  
for your program to read.

### CAUTION!!!



**NOTE: Your submitted file(s) MUST be spelled and cased as instructed.**