/\*

1. Obtain first names, last names, and 3 integer

test scores for each of 12 students

Store in an array named records

Use a PCF named get\_records( )

Use data file /home/faculty/urbanc/ghp289.dat

The data file is composed as follows:

column 0: first name

column 1: last name

column 2: test score #1

column 3: test score #2

column 4: test score #3

Each row represents a single student.

2. Determine the test score average for each

student

Store in an array named records

Develop and use a PCF named compute\_avgs( )

3. Determine the letter grade for each student

Store in an array named records

Use this scale:

>= 90 is an A

>= 80 and less than 90 is a B

>= 70 and less than 80 is a C

>= 65 and less than 70 is a D

< 65 is an F

Develop and use a PCF named determine\_grades( )

4. Print the first name, last name, and average test

score and the letter grader for each of the 10 students

Develop and use a PCF named display( ).

5. Terminate

\*/

#include <stdio.h>

//Structure definition below

typedef struct student { char first\_name[15];

char last\_name[15];

int test1;

int test2;

int test3;

float average;

char letter\_grade;

} student ;

//PCF Prototypes below

void get\_records(student[12], int);

void compute\_avg(student[12], int);

void determine\_grades(student[12], int);

void display(student[12], int);

int main ( void )

{

//Local variables created below

student records[12];

int rows = 12;

//1. Obtain first names, last names, and 3 integer

// test scores for each of 12 students

// Store in an array named records

// Use a PCF named get\_records( )

// Use data file /home/faculty/urbanc/ghp289.dat

// The data file is composed as follows:

// column 0: first name

// column 1: last name

// column 2: test score #1

// column 3: test score #2

// column 4: test score #3

// Each row represents a single student.

get\_records(records, rows);

//2. Determine the test score average for each

// student

// Store in an array named records

// Develop and use a PCF named compute\_avgs( )

compute\_avg(records, rows);

//3. Determine the letter grade for each student

// Store in an array named records

// Use this scale:

// >= 90 is an A

// >= 80 and less than 90 is a B

// >= 70 and less than 80 is a C

// >= 65 and less than 70 is a D

// < 65 is an F

// Develop and use a PCF named determine\_grades( )

determine\_grades(records, rows);

//4. Print the first name, last name, and average test

// score and the letter grader for each of the 10 students

// Develop and use a PCF named display( ).

display(records, rows);

//5. Terminate

return ( 0 ) ;

}

//PCF Definitions below

void get\_records(student records[12], int r)

{

FILE \* read\_ptr = NULL ;

//local variables declared below

int rows =0 ;

read\_ptr = fopen("/home/faculty/urbanc/ghp289.dat" , "r");

if (read\_ptr == NULL)

{

printf("\n\nghp289.dat not opened.\n\n") ;

}

else

{

printf("\n\nghp289.dat opened properly.\n\n") ;

}

for(rows = 0; rows < r; rows ++)

{

fscanf(read\_ptr,"%s", records[rows].first\_name);

fscanf(read\_ptr,"%s", records[rows].last\_name);

fscanf(read\_ptr,"%d", &records[rows].test1);

fscanf(read\_ptr,"%d", &records[rows].test2);

fscanf(read\_ptr,"%d", &records[rows].test3);

}

fclose(read\_ptr);

printf("\n"); //line spacing

return;

}

void compute\_avg(student records[12], int r)

{

//local variables declared below

int rows = 0, sum = 0;

for(rows = 0; rows < r; rows ++)

{

sum = records[rows].test1 + records[rows].test2

+ records[rows].test3 ;

records[rows].average = sum / 3.0 ;

sum = 0;

}

return;

}

void determine\_grades(student records[12], int r)

{

//local variables declared below

int rows = 0;

for(rows = 0; rows < r; rows ++)

{

if(records[rows].average >= 90)

{

records[rows].letter\_grade = 'A';

}

else if(records[rows].average >= 80)

{

records[rows].letter\_grade = 'B';

}

else if(records[rows].average >= 70)

{

records[rows].letter\_grade = 'C';

}

else if(records[rows].average >= 65)

{

records[rows].letter\_grade = 'D';

}

else

{

records[rows].letter\_grade = 'F';

}

}

return;

}

void display(student records[12], int r)

{

//local variables declared below

int rows = 0;

for(rows = 0; rows < r; rows ++)

{

printf("%s %s %.2f %c \n", records[rows].first\_name,

records[rows].last\_name, records[rows].average,

records[rows].letter\_grade);

}

printf("\n"); //line spacing

return;

}