Write a program (CardOrdering) that reads the suit and the face of cards for a hand from a file (Cards.txt) into an array of structures and then orders the cards with respect to suits and faces from that, then it writes them to an output file. Use the following include file (Card.h) for the definition of the Card structure and the associated helper functions to display the suit and the number of a card.

/\*

This header file contains the definitions for the CardOrdering program.

\*/

#define SUCCESS\_EXIT 0

#define ERROR\_EXIT -1

#define RANDOM\_CARD\_FILE "RandomCards.txt"

#define ORDERED\_CARD\_FILE "OrderedCards.txt"

#define MAX\_STACK\_SIZE 13

void DisplaySuit(FILE\* fptr, unsigned char suit); // prints the suit as text

void DisplayNumber(FILE\* fptr, unsigned short item); // prints the item as number text

void DisplayFace(FILE\* fptr, unsigned short item); // prints the item as face text

typedef struct card

{

unsigned char suit; // 'S' for Spade, 'H' for Heart, 'D' for Diamond and 'C' for Club

unsigned short item; // 1-10 for numbers and 11-13 for Jack, Queen and King

void (\*ptrDisplayItem) (FILE\* fptr, unsigned short item); // pointer to display number or face

} Card;

Make sure the solution includes the following 4 files:

1. Card.h: The contents of this file is shown above.
2. Card.c: This file contains the implementation of the helper functions.
3. CardOrdering.c: This file contains the implementation of the ‘main’ function to open and read the cards from the Cards.txt file into and array of Card structures, orders them with respect to suits and numbers/faces and writes them to the ouput file. Set the function pointer ptrDisplayItem in the structure to call the DisplayNumber and DisplayFace functions.
4. Cards.txt: Each line contains the suit and the face of a card. The contents of a sample file is shown below.

S 5

H 3

D 5

S 2

H 4

D 11

H 5

C 15

G 3

S 0

H 12

D 9

C 5

(Note: Yes, it is a silly application to get you familiar with real life applications!..)