

```

//*****
//
//      sample solution for pa10 -- Fall 2005
//
//*****
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>
using namespace std;

// constants for array dimensions

const int ROWS = 10, COLUMNS = 15;

// function prototypes

void Init(char[][COLUMNS]);
void SellSeats(double[],char[][COLUMNS]);
void Display(double[],char[][COLUMNS]);
int TotalSeats(char[][COLUMNS],char);
double TotalSales(double[],char[][COLUMNS]);

int main()
{
    // declare all necessary variables
    int      i, row, choice, totalSeatsS, totalSeatsA;
    char     chart[ROWS][COLUMNS];
    double   price, prices[ROWS], totalSales;
    ifstream inFile;

    // setup printing of floating point numbers (it's money -- so 2 decimal places)
    cout << fixed << showpoint;
    cout.precision(2);

    // read in price data before beginning menu system
    inFile.open("prices.dat");
    for(i=0;i<ROWS;i++)
    {
        inFile>>row>>price;
        prices[row] = price;
    }
    inFile.close();

    do
    {
        cout << "\n\n1. Initialize All Seats to Available\n";
        cout << "2. Sell Seats\n";
        cout << "3. Display Seating Chart\n";
        cout << "4. Display Total Seats Sold and Total Seats Available\n";
        cout << "5. Display Total Ticket Sales\n";
        cout << "6. Quit\n\n";

        cout << "Selection: ";
        cin >> choice;
    }
}

```

```

switch(choice)
{
    case 1: Init(chart);
            break;

    case 2: SellSeats(prices,chart);
            break;

    case 3: Display(prices,chart);
            break;

    case 4: totalSeatsS = TotalSeats(chart,'S');
            totalSeatsA = TotalSeats(chart,'*');
            cout << "\n    Total Seats Sold: " << setw(4) << totalSeatsS << endl;
            cout << "Total Seats Available: " << setw(4) << totalSeatsA << endl;
            break;

    case 5: totalSales = TotalSales(prices,chart);
            cout << "\nTotal Sales: $" << setw(4) << totalSales << endl;
            break;

    case 6: cout << "Thanks -- Goodbye !!\n\n";
            break;

    default: cout << "Invalid Choice -- Try Again !!\n\n";
            break;
}

} while (choice != 6);

return 0;
}

```

```

//***** Function Definitions *****

```

```

void Init(char chart[][COLUMNS])
{
    int i,j;

    for(i=0;i<ROWS;i++)
        for(j=0;j<COLUMNS;j++)
            chart[i][j] = '*';
}

```

```

void SellSeats(double prices[], char chart[][COLUMNS])
{
    int    i, row, beginSeat, endSeat, numSeats;
    bool   goodSale;
    double totalPrice;

    cout << "Please enter the row number: ";
    cin >> row;
    cout << "Please enter the beginning seat number: ";
    cin >> beginSeat;
    cout << "Please enter the ending seat number: ";
    cin >> endSeat;
}

```

```

// adjust for array bounds beginning at zero (instead of 1)
row--;
beginSeat--;
endSeat--;

goodSale = true; // start off with the sale being "possibly" good/true

for(i=beginSeat;i<=endSeat;i++)
    if (chart[row][i] == 'S') // if, at any point, a seat is already
        goodSale = false; // sold, change goodSale to false

if (goodSale == true)
{
    for(i=beginSeat;i<=endSeat;i++) // mark all seats sold 'S'
        chart[row][i] = 'S';

    numSeats = endSeat - beginSeat + 1; // determine price of seats just sold
    totalPrice = prices[row] * numSeats;

    cout << "\n\nCongratulations -- These seats are available!\n";
    cout << "Total Price: $ " << totalPrice << "\n\n";
}
else
    cout << "\n\nSorry -- this block of seats is unavailable!\n\n";
}

void Display(double prices[], char chart[][COLUMNS])
{
    int i,j;

    for(i=0;i<ROWS;i++)
    {
        for(j=0;j<COLUMNS;j++)
        {
            cout << setw(3) << chart[i][j];
        }
        cout << endl;
    }
}

int TotalSeats(char chart[][COLUMNS], char symbol)
{
    int i, j, count=0;

    for(i=0;i<ROWS;i++)
        for(j=0;j<COLUMNS;j++)
            if (chart[i][j] == symbol) // symbol is either a '*' or a 'S'
                count++;

    return count;
}

```

```
double TotalSales(double prices[], char chart[][COLUMNS])
{
    int    i, j, count;
    double sum=0;

    for(i=0;i<ROWS;i++)
    {
        count = 0;           // count for tickets solds on this (i) row
        for(j=0;j<COLUMNS;j++)
        {
            if (chart[i][j] == 'S')
                count++;
        }
        sum = sum + (count * prices[i]); // add in ticket sales for this (i) row
    }

    return sum;
}
```