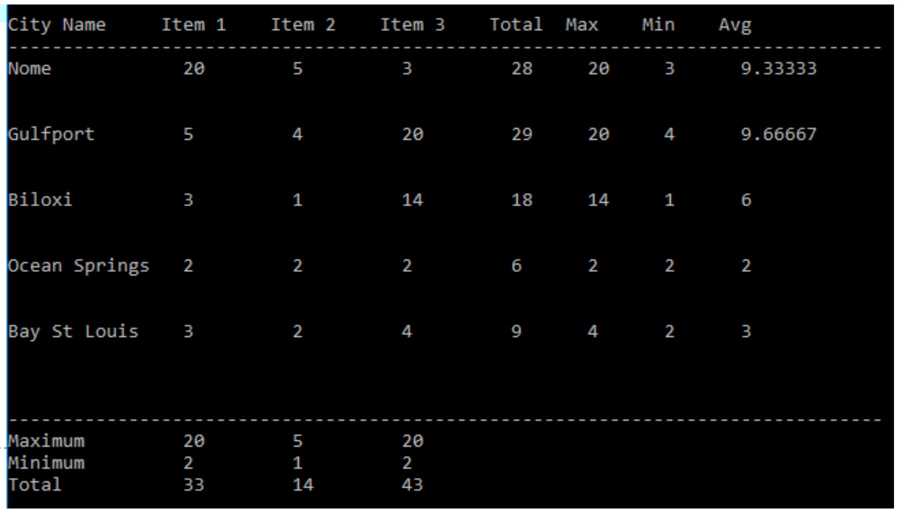
I and 2 dimensional arrays

```
using namespace std;
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>

void printline();

int main()
{ string name[5];

  int sales[5][3],
  totalrow[5],
  maxrow[5],
  minrow[5],
  totalcol[3],
  maxcol[3],
  mincol[3];
  double average[5];

  char graph[5][40];

  int i, j, k, l, max;

//-----------------------------------------------------------------------
// Read from datafile into arrays

    ifstream fin;
    fin.open("sales.dat");
    cout <<"\n\n";
    cout <<setw(14)<<left<<"City Name"<<setw(10)<<"Item 1"<<setw(10)<<"Item 2"<<setw(10)<<"Item 3"<<setw(7)<<"Total"<<setw(7)<<"Max"<<setw(7)<<"Min" <<setw(7)<<"Avg";
    printline();

    for (i=0;i<3;i++)  // Initialize column accumulators
      {maxcol[i]=0;
       mincol[i]=20000;
       totalcol[i]=0;
      }
```



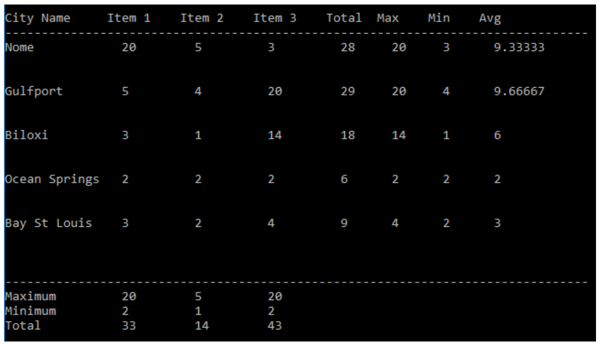
```
City Name     Item 1    Item 2    Item 3    Total  Max    Min    Avg
--------------------------------------------------------------------------------
Nome            20        5         3         28     20     3      9.33333


Gulfport        5         4         20        29     20     4      9.66667


Biloxi          3         1         14        18     14     1      6


Ocean Springs   2         2         2         6      2      2      2


Bay St Louis    3         2         4         9      4      2      3



--------------------------------------------------------------------------------
Maximum         20        5         20
Minimum         2         1         2
Total           33        14        43
```

Initializing the total, max and min for the column ***j*** before the nested loop.

```
    //--------------------------------------------------

    for (i = 0; i<5; i++)
    {   getline(fin,name[i]);
        cout <<setw(16)<<left<< name[i];

// Initialize row accumulators

        totalrow[i]=0;
        maxrow[i]=-1;
        minrow[i]=100000;

        for (j = 0; j<3; j++)    // Read in the sales for each location in a loop
        {
            fin >> sales[i][j];

// This section check if the new value is either the new maximum or minimum for the row and the column

            if (sales[i][j]>maxrow[i])
               maxrow[i]=sales[i][j];
            if (sales[i][j]<minrow[i])
               minrow[i]=sales[i][j];
            if (sales[i][j]>maxcol[j])
               maxcol[j]=sales[i][j];
            if (sales[i][j]<mincol[j])
               mincol[j]=sales[i][j];

// This section accumulates the new sales for the totals of the row and column

            totalrow[i]=totalrow[i]+sales[i][j];
            totalcol[j]=totalcol[j]+sales[i][j];

            cout << setw(10)<<sales[i][j];
        }
       average[i]=totalrow[i]/3.0;
       cout <<setw(7)<< totalrow[i] <<setw(7)<<maxrow[i] <<setw(7) <<minrow[i]<<setw(7) <<average[i]<< endl;
       fin.ignore();
       cout <<"\n\n";
    }
```

Initializing the total, max and min for the row ***I***.

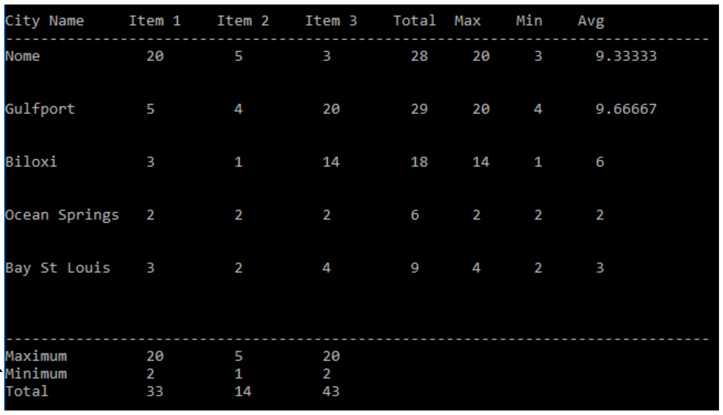
Checking the new sales value to see if it is the new max or min for row ***I*** and column ***j*** .



```
City Name     Item 1    Item 2    Item 3    Total  Max    Min    Avg
--------------------------------------------------------------------------------
Nome            20        5         3         28     20     3      9.33333


Gulfport        5         4         20        29     20     4      9.66667


Biloxi          3         1         14        18     14     1      6


Ocean Springs   2         2         2         6      2      2      2


Bay St Louis    3         2         4         9      4      2      3



--------------------------------------------------------------------------------
Maximum         20        5         20
Minimum         2         1         2
Total           33        14        43
```

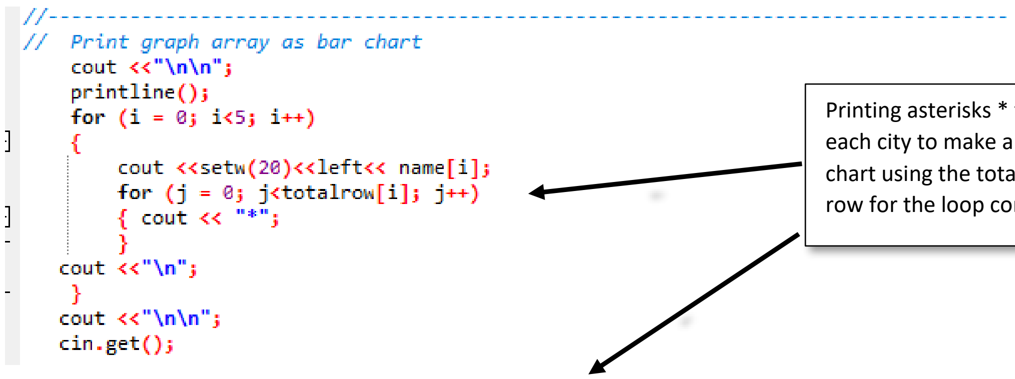
```
    printline();
    cout <<setw(16)<<left<<"Maximum";
    for (i=0;i<3;i++)
      cout << setw(10)<<maxcol[i];

    cout <<"\n";
    cout <<setw(16)<<left<<"Minimum";
    for (i=0;i<3;i++)
      cout<< setw(10) <<mincol[i];
    cout <<"\n";

    cout <<setw(16)<<left<<"Total";
    for (i=0;i<3;i++)
      cout<< setw(10) <<totalcol[i];
    cout <<"\n";

    cin.get();
```

Printing the maximum, minimum and total of the columns after the exit of the nested loops.



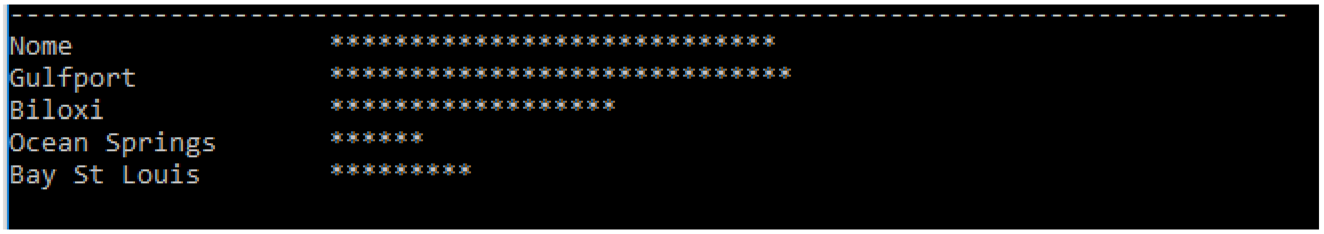
```
City Name     Item 1    Item 2    Item 3    Total  Max    Min    Avg
--------------------------------------------------------------------------------
Nome            20        5         3         28     20     3      9.33333


Gulfport        5         4         20        29     20     4      9.66667


Biloxi          3         1         14        18     14     1      6


Ocean Springs   2         2         2         6      2      2      2


Bay St Louis    3         2         4         9      4      2      3



--------------------------------------------------------------------------------
Maximum         20        5         20
Minimum         2         1         2
Total           33        14        43
```

Charts

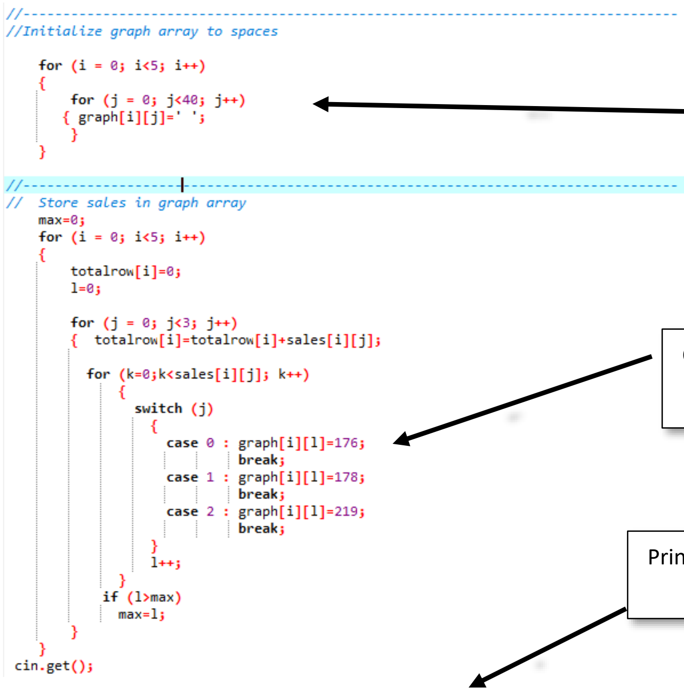


```
//-----------------------------------------------------------------------------
//  Print graph array as bar chart
    cout <<"\n\n";
    printline();
    for (i = 0; i<5; i++)
    {
        cout <<setw(20)<<left<< name[i];
        for (j = 0; j<totalrow[i]; j++)
        { cout << "*";
        }
   cout <<"\n";
    }
   cout <<"\n\n";
   cin.get();
```

Printing asterisks * for each city to make a bar chart using the total of the row for the loop control.



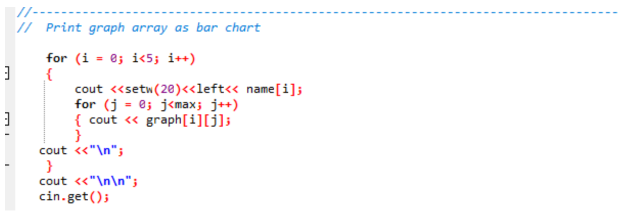
```
--------------------------------------------------------------------------------
Nome                ****************************
Gulfport            *****************************
Biloxi              ******************
Ocean Springs       ******
Bay St Louis        *********
```



```
//--------------------------------------------------------------------------
//Initialize graph array to spaces

    for (i = 0; i<5; i++)
    {
        for (j = 0; j<40; j++)
       { graph[i][j]=' ';
        }
    }

//--------------------------------------------------------------------------
//  Store sales in graph array
    max=0;
    for (i = 0; i<5; i++)
    {
        totalrow[i]=0;
        l=0;

        for (j = 0; j<3; j++)
        {  totalrow[i]=totalrow[i]+sales[i][j];

          for (k=0;k<sales[i][j]; k++)
            {
              switch (j)
               {
                 case 0 : graph[i][l]=176;
                          break;
                 case 1 : graph[i][l]=178;
                          break;
                 case 2 : graph[i][l]=219;
                          break;
               }
               l++;
            }
           if (l>max)
             max=l;
        }
    }
 cin.get();
```

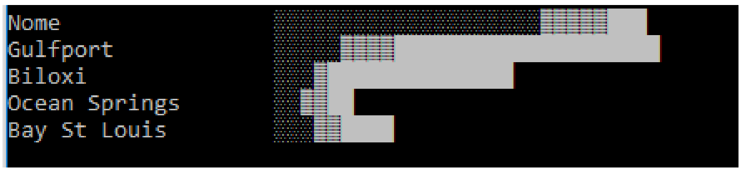
Initializing the array to spaces to create stacked column charts. Needed since the graph will be rotated for a stacked column bar chart.

Creating a bar chart using ASCII symbols.

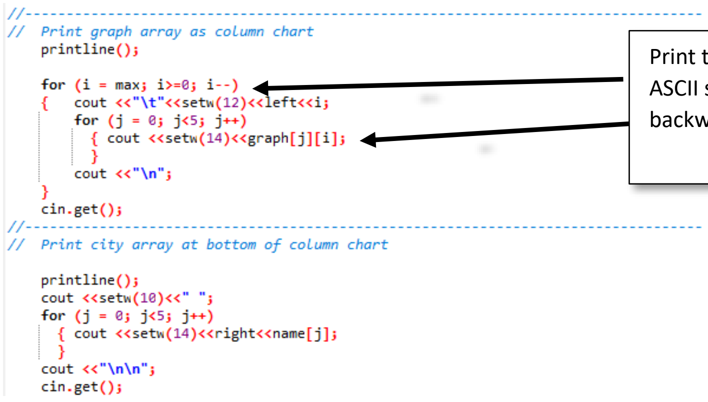


```
//--------------------------------------------------------------------------
//  Print graph array as bar chart

     for (i = 0; i<5; i++)
     {
         cout <<setw(20)<<left<< name[i];
         for (j = 0; j<max; j++)
         { cout << graph[i][j];
         }
    cout <<"\n";
     }
    cout <<"\n\n";
    cin.get();
```

Print the bar chart using ASCII symbols.

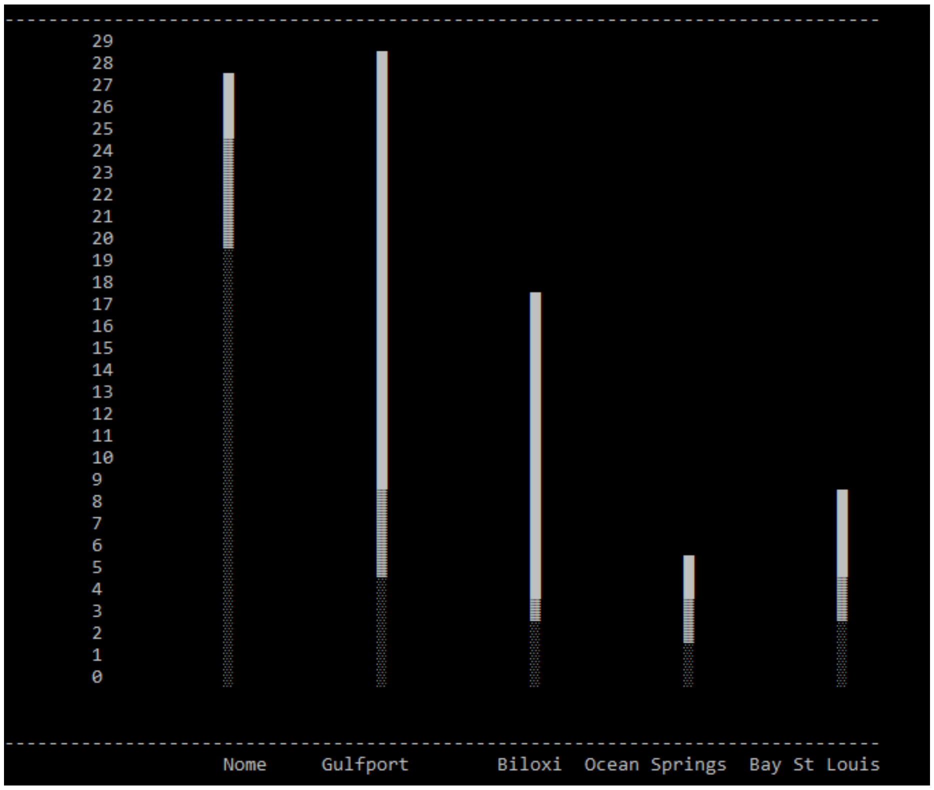


```
Nome
Gulfport
Biloxi
Ocean Springs
Bay St Louis
```



```
//----------------------------------------------------------------------------------
//  Print graph array as column chart
    printline();

    for (i = max; i>=0; i--)
    {   cout <<"\t"<<setw(12)<<left<<i;
        for (j = 0; j<5; j++)
          { cout <<setw(14)<<graph[j][i];
          }
        cout <<"\n";
    }
    cin.get();
//----------------------------------------------------------------------------------
//  Print city array at bottom of column chart

    printline();
    cout <<setw(10)<<" ";
    for (j = 0; j<5; j++)
      { cout <<setw(14)<<right<<name[j];
      }
    cout <<"\n\n";
    cin.get();
```

Print the graph as a stacked column chart using ASCII symbols by printing the rows (counter i) backwards the graph [j] [I].



The files can be downloaded Sales1 Program   Sales Datafile