CSCI 2170 **Lecture Notes (1)**
Review

- **Array**
  - **1-D array**

```
for (i=0; i<numItems; i++)
        cin >> array[i];
```

read in a list of integer values that ends with a sentinel value (from keyboard or file)

how many values are read?
What is the average of these values?
Is a particular value in the list?
[sort all values in non-descending order]
[find number of values that are greater than its previous value]

  - **2-D array**

```
 for (int i=0; i<numOfRows; i++)
        for (int j=0; j<numOfCols; j++)
                cin>> array[i][j];
```

Read in a maze from a file and store the maze in a 2-D array of character
Find the column averages and/or row averages of a 2-D array of student test scores

  - **Parallel array**

Need to store information about which movie theater is currently showing which movies. Three blocks of information:
Group 1:
For each movie, information about title, genre, running time, director, year made, … ➔ assuming we only need to keep track of the title information.

Array 1 :	movies
0	“Munich”/R/204/
1	“Harry Potter and the Goblet of File”/PG13/140/
2	“King Kong”/PG13/180/
3	…
…	
numMovies-1	“Casanova”/R/108/

define the struct type MovieType:

```
enum  GenreType {R, PG, PG13};
struct movieType
```

```
{          string  title;
           GenreType  genre;
           int  time;
};

const int MAX_MOVIES = 25;
MovieType    movies[MAX_MOVIES];
```

Group 2:

For each movie theater, information about the name of the theater, the phone number, the company, the address, type,

```
     Array 2:      theaters
           0          "Opry Mills 20"/343-9844/Regal/
           1          "Hollywood 27"/837-2322/Regal/
           2          "Bellevue 8"/883-4444/Carmike/
           3          "Hickory 8"/234-2232/Carmike/
                     …
numTheaters-1       "Thorouboug"/454-3289/Regal/
```

```
struct TheaterType
{
           string   name;
           string   phone;
           string   company;
};
const int MAX_THEATERS=30;
TheaterType     theaters[MAX_THEATERS];
```

Group 3:

For each movie theater, it is showing a list of movies

```
     Array 3:      showings (index of movie in array "movies"
           0          1, 2, 4, -1, -1, -1
           1          0, 5, -1, -1, -1
           2          1, 2, 7, 8, -1, -1, -1
           3          3, 4, 9, -1, -1, …
                     …
numTheaters-1       1, 3, 4, 7, -1, -1, …
```

```
int showings[MAX_THEATERS][MAX_MOVIES];
// initialize all elements of the 2D array to -1

// read in the showings at each theater as an index value corresponds to the
// movie in array "movies"
```

NOW, you can search for information about these theaters/movies

- Show all theaters who are currently showing movie

  ```
  title = “King Kong” ;

  cout << “The following theaters are currently showing “ <<
  title << “:”<< endl;

  for (i=0; i<numTheaters; i++)
  {
     for (j=0; showings[i][j] >=0; j++)
     {
             if (movies[showings[i][j]].title ==title)
                cout << theaters[i].name << endl;
      }
  }
  ```

- List all movies current shown at the specified theater “theater”.

  ```
  int i, j;
  theater = “Hickory 8”;
  for (i=0; i<numTheatures; i++)
  {
      if (theaters[i].name == theater)
             break;
  }
  // i  is equal to the index of the “theater” in the theater-list
  cout << “These are the list of movies currently showing at “
  << theater << endl;
  for (j=0; showings[i][j]>=0; j++)
      cout << movies[showings[i][j]].title << “ “ << endl;
  ```

- List all PG13 movies currently being shown in movie theaters
- List all PG13 movies currently shown in theater “Hickory 8”
- List all movies that are shown both in theater “Hickory 8” and “Hollywood 20”
- Print the titles of all movies shown in theater “Hickory 8”

- **Command line arguments (C-type string)**

Used to provide information to the program using command line arguments.

Provides 3 values at command line, and compute and display the sum of the three values
Provide 2 values at command line, and indicate whether the first value is less than, equal to, or greater than the second value.

```
frank%    a.out    movie.dat   10   movie.report
```

In this case, there are 3 command line arguments → argc ==3
The three arguments are stored in the 2-d array of character “argv”:

argv[0] = “a.out”
argv[1] = “movie.dat”
argv[2] = “movie.report”

Each of the 3 argv’s are C-type string

```
int main(int argc, char *argv[])
{
    string        titles[MAX_TITLES];
    int           numberOfTitles;
    ifstream      infile;
    ofstream      outfile;

     if (argc <= 4)
     {
          cerr << “Incorrect number of command line arguments.”;
          cout << “Run the program with :“
                << “a.out    input-file   number-of-movies   output-file”<< endl;
          exit(-1);
      }
     else
    {
          infile.open(argv[1]);
          assert(infile);

          numberOfTitles = atoi(argv[2]);  // atoi() defined in <cstdlib>
          for (int i=0; i<numOfTitles && numOfTitles < MAX_TITLES; i++)
         {
                    getline(infile, titles[i]);
          }

          outfile.open(argv[3]);
          assert(outfile);

          ….  // write  movie info to output file
    }

    …
    return 0;
}
```