Even Semester (2021)

BINUS UNIVERSITY
BINUS INTERNATIONAL

Student Information:

	Surname	Given Names	Student ID Number
1.	**Yamin**	**Farras**	**2101704584**
2.	**Nofarditya Ashadi**	**Arkaan**	**2101718425**
3.	**Christopher Chandra**	**Jason**	**2101725033**

Course Code	**: COMP6048**	**Course Name**	**: Data Structures**
Class	**: L2BC**	**Name of Lecturer(s)**	: 1. Raymondus Raymond Kosala 2. Maria Seraphina Astriani
Major	**: CS**		

Title of Assignment :
(if any)

Type of Assignment : Final Project

Submission Pattern

Due Date	**: 4-06-2018**	**Submission Date**	**:**

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Movie Sorter

Problem

For our Data Structures Final Project, we made a program for sorting movies.

We tested a few types of data structures to find out which one is the most efficient.

Goal

To find the most efficient method to sort movies.

Alternative Data Structures

Originally, we chose to use arrays as the data structure for the sorting program but we found out that if we used arrays, the list would not have been dynamic. So, we used Vectors.

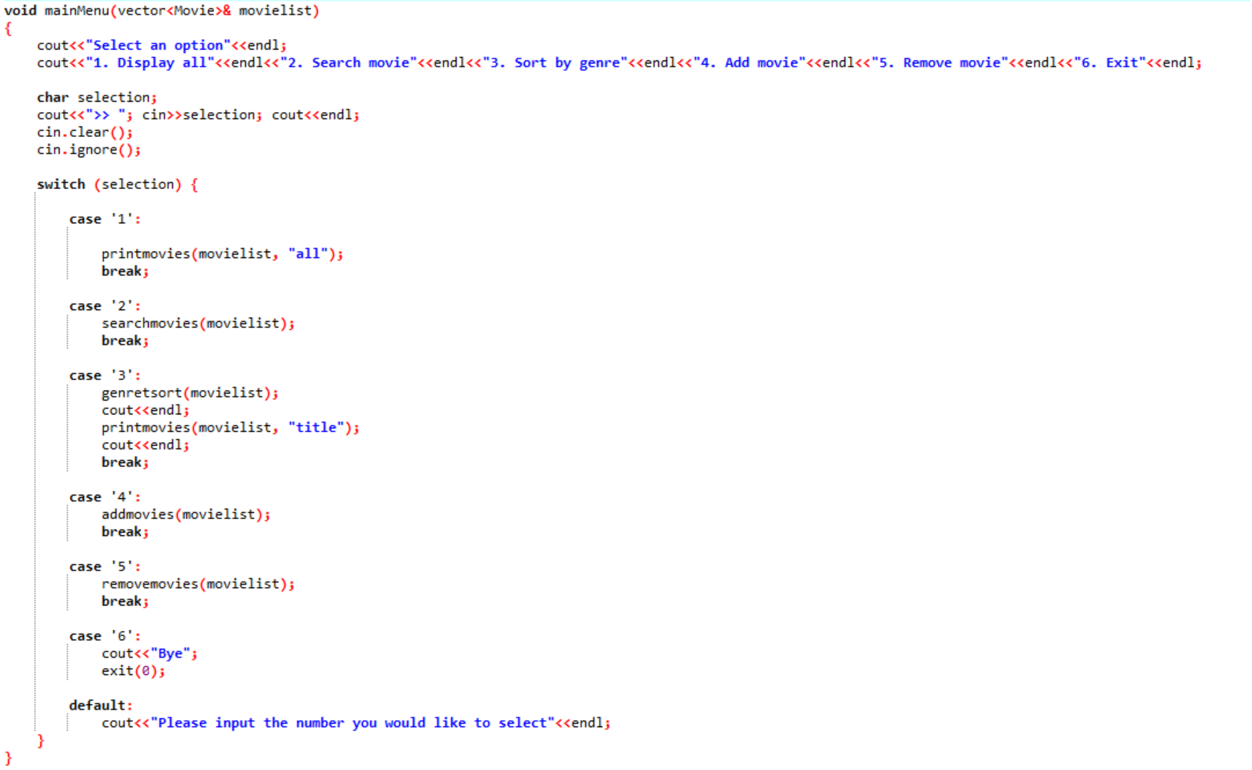
Theoretical Analysis

We use Vector over Array because:

1. **Synchronization :** Vector is synchronized that means at a time only one thread can access the code while array is not synchronized that means multiple threads can work on array at same time.
2. **Performance:** Array is faster as it is non-synchronized while vector operations give slow performance as they are synchronized(thread-safe). If one thread works on vector has acquired lock on it which makes other thread will has to wait till lock is released.
3. **Data Growth:** Array and Vector both grow and shrink dynamically to maintain optimal use of storage. But the way they resized is different. Array increments 50% of current array size if number of elements exceeds its capacity while vector increments 100% means doubles the current array size.
4. **Traversal:** Vector can use both Enumeration and Iterator for traversing over elements of vector while array can only use Iterator for traversing.

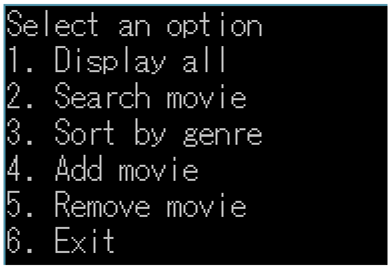
Manual and Result

1. Main Menu



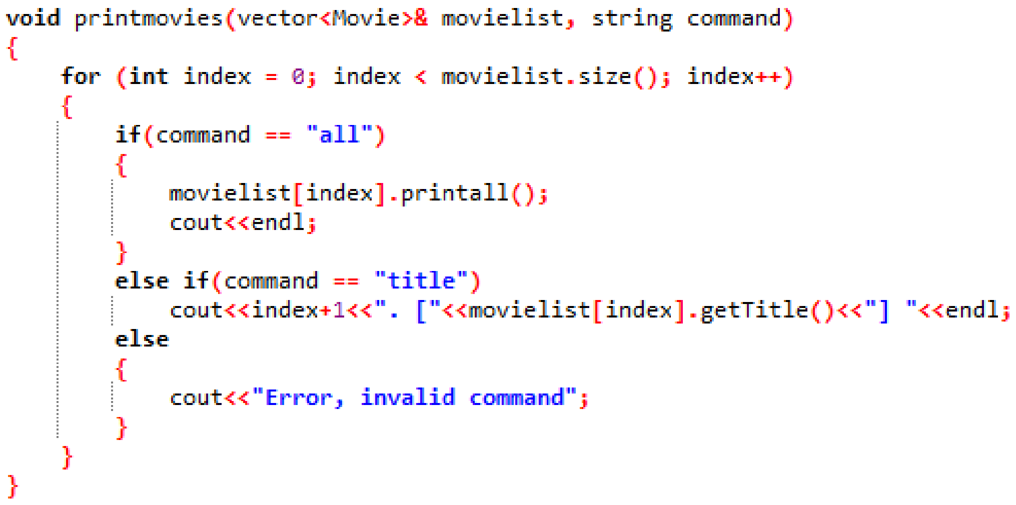
```cpp
void mainMenu(vector<Movie>& movielist)
{
    cout<<"Select an option"<<endl;
    cout<<"1. Display all"<<endl<<"2. Search movie"<<endl<<"3. Sort by genre"<<endl<<"4. Add movie"<<endl<<"5. Remove movie"<<endl<<"6. Exit"<<endl;

    char selection;
    cout<<">> "; cin>>selection; cout<<endl;
    cin.clear();
    cin.ignore();

    switch (selection) {

        case '1':

            printmovies(movielist, "all");
            break;

        case '2':
            searchmovies(movielist);
            break;

        case '3':
            genretsort(movielist);
            cout<<endl;
            printmovies(movielist, "title");
            cout<<endl;
            break;

        case '4':
            addmovies(movielist);
            break;

        case '5':
            removemovies(movielist);
            break;

        case '6':
            cout<<"Bye";
            exit(0);

        default:
            cout<<"Please input the number you would like to select"<<endl;
    }
}
```

mainMenu function will show six options, which are display all movies, search certain movie, sort movies by genre, add new movie, remove existing movie, and exit the program. If you input besides 1-6, it will return “Please input the number you would like to select”.



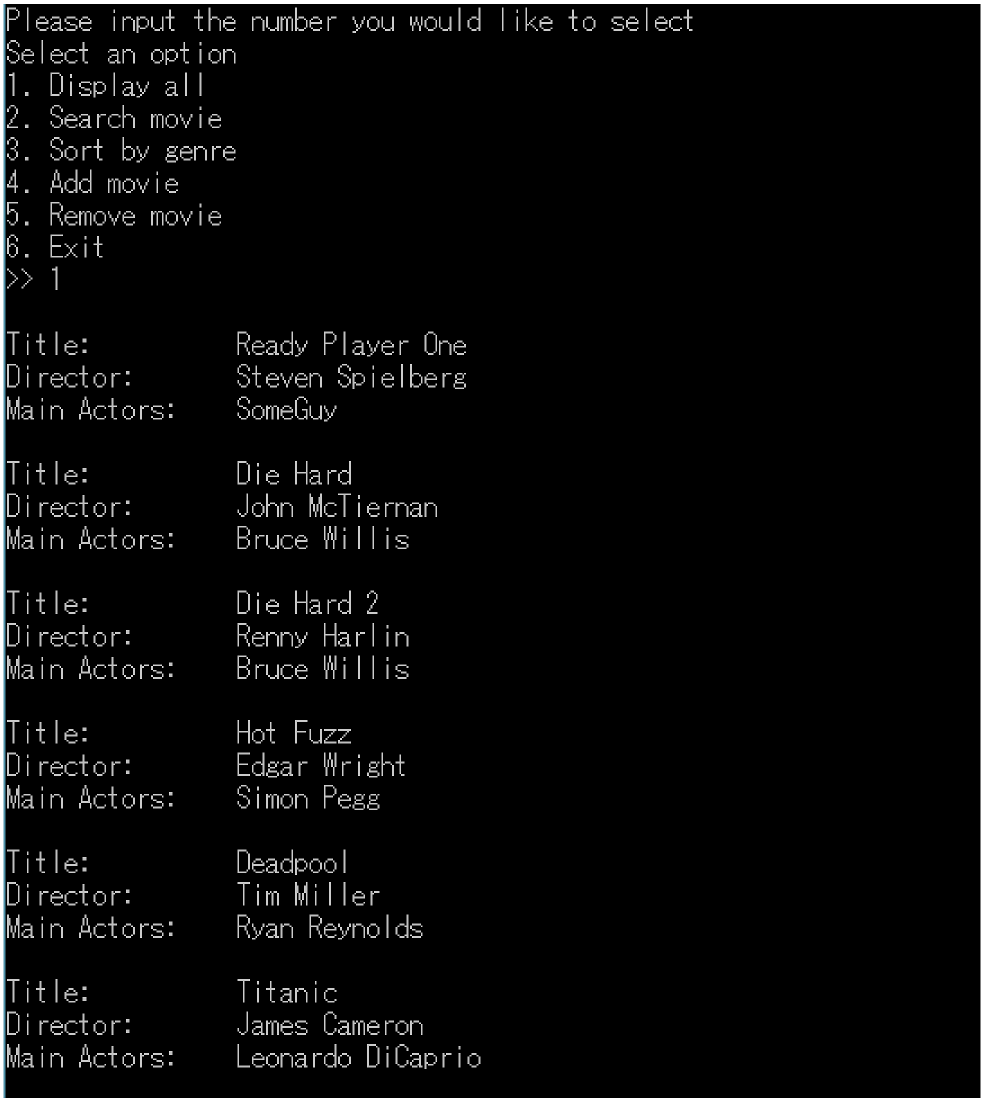
```
Select an option
1. Display all
2. Search movie
3. Sort by genre
4. Add movie
5. Remove movie
6. Exit
```

2. Display all



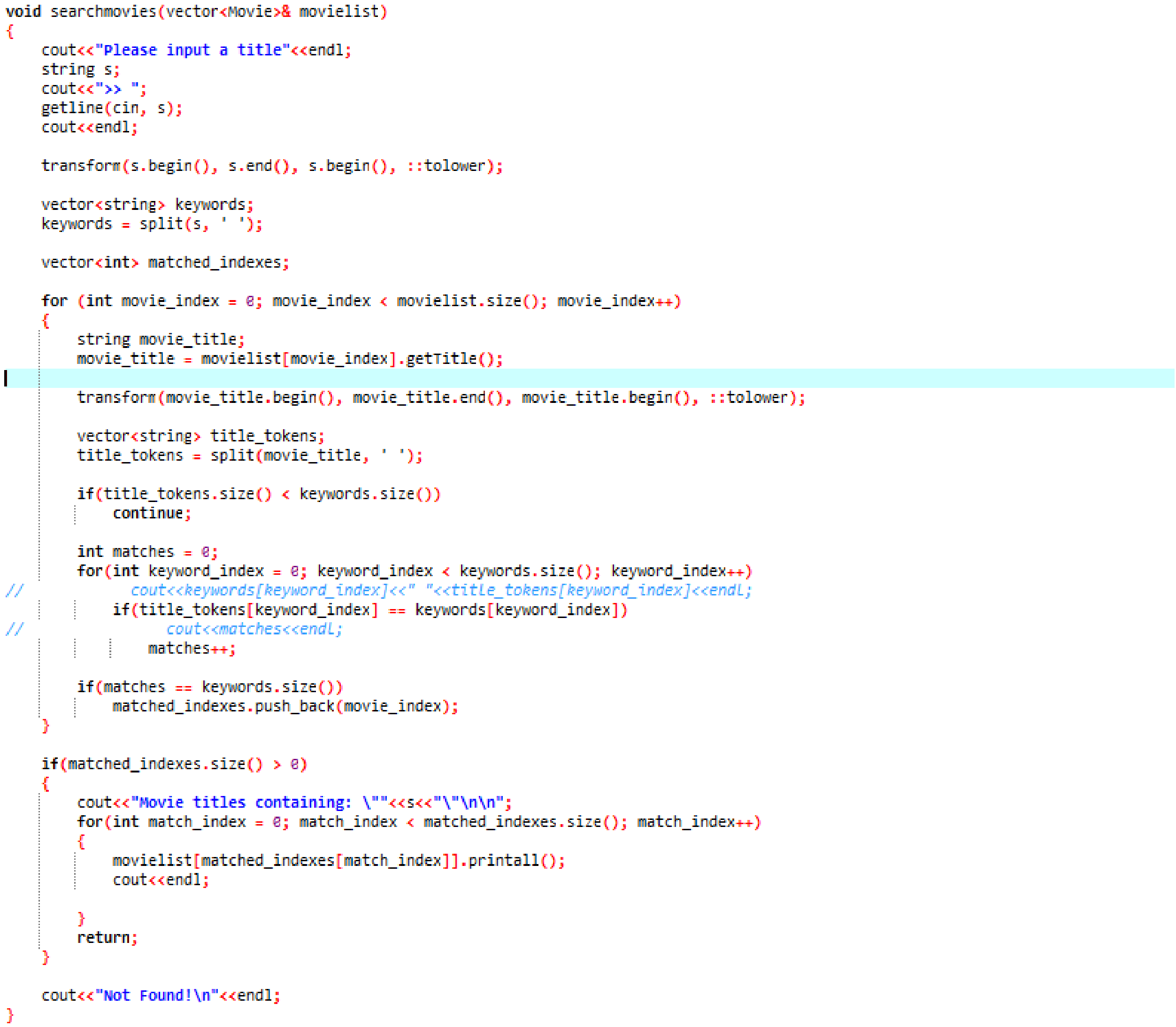
```
void printmovies(vector<Movie>& movielist, string command)
{
    for (int index = 0; index < movielist.size(); index++)
    {
        if(command == "all")
        {
            movielist[index].printall();
            cout<<endl;
        }
        else if(command == "title")
            cout<<index+1<<". ["<<movielist[index].getTitle()<<"] "<<endl;
        else
        {
            cout<<"Error, invalid command";
        }
    }
}
```

If you input 1, it will display all movies that we have and details about each movie such as title, director, and main actor.



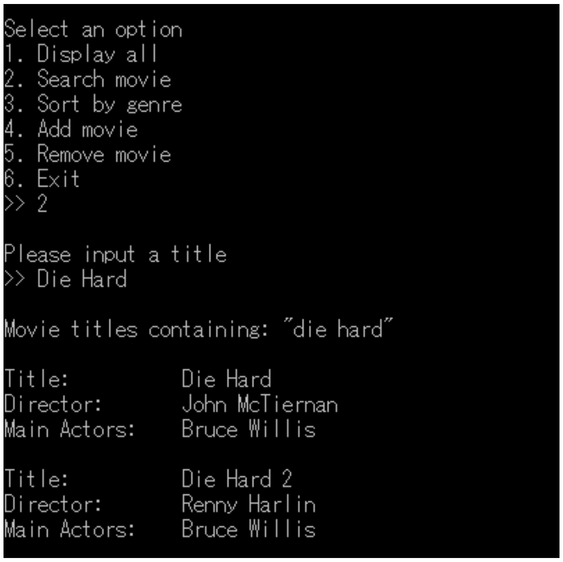
```
Please input the number you would like to select
Select an option
1. Display all
2. Search movie
3. Sort by genre
4. Add movie
5. Remove movie
6. Exit
>> 1

Title:          Ready Player One
Director:       Steven Spielberg
Main Actors:    SomeGuy

Title:          Die Hard
Director:       John McTiernan
Main Actors:    Bruce Willis

Title:          Die Hard 2
Director:       Renny Harlin
Main Actors:    Bruce Willis

Title:          Hot Fuzz
Director:       Edgar Wright
Main Actors:    Simon Pegg

Title:          Deadpool
Director:       Tim Miller
Main Actors:    Ryan Reynolds

Title:          Titanic
Director:       James Cameron
Main Actors:    Leonardo DiCaprio
```

3. Search Movie



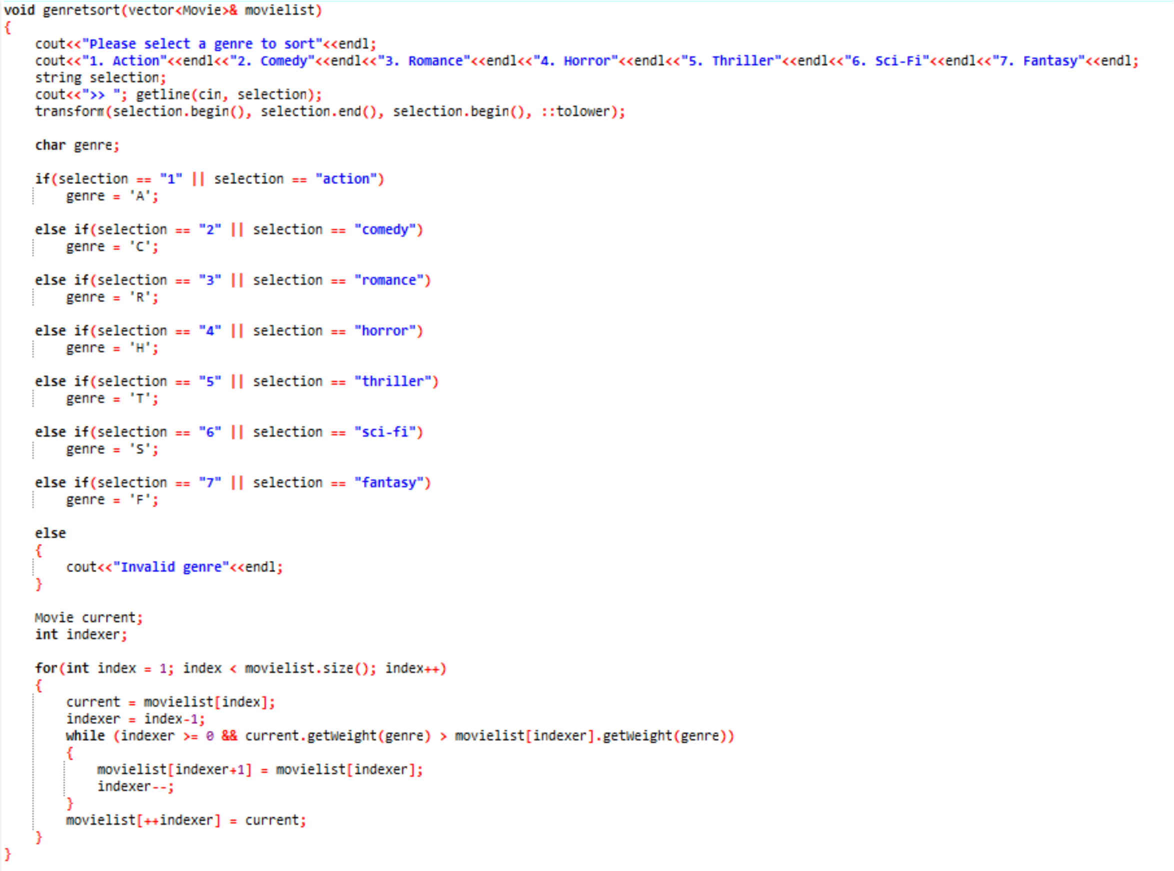
```
void searchmovies(vector<Movie>& movielist)
{
    cout<<"Please input a title"<<endl;
    string s;
    cout<<">> ";
    getline(cin, s);
    cout<<endl;

    transform(s.begin(), s.end(), s.begin(), ::tolower);

    vector<string> keywords;
    keywords = split(s, ' ');

    vector<int> matched_indexes;

    for (int movie_index = 0; movie_index < movielist.size(); movie_index++)
    {
        string movie_title;
        movie_title = movielist[movie_index].getTitle();

        transform(movie_title.begin(), movie_title.end(), movie_title.begin(), ::tolower);

        vector<string> title_tokens;
        title_tokens = split(movie_title, ' ');

        if(title_tokens.size() < keywords.size())
            continue;

        int matches = 0;
        for(int keyword_index = 0; keyword_index < keywords.size(); keyword_index++)
//            cout<<keywords[keyword_index]<<" "<<title_tokens[keyword_index]<<endl;
            if(title_tokens[keyword_index] == keywords[keyword_index])
//                cout<<matches<<endl;
                matches++;

        if(matches == keywords.size())
            matched_indexes.push_back(movie_index);
    }

    if(matched_indexes.size() > 0)
    {
        cout<<"Movie titles containing: \""<<s<<"\"\n\n";
        for(int match_index = 0; match_index < matched_indexes.size(); match_index++)
        {
            movielist[matched_indexes[match_index]].printall();
            cout<<endl;

        }
        return;
    }

    cout<<"Not Found!\n"<<endl;
}
```

If you input 2, you can search certain movie. You can input the exact title and you will get the movie you want, or you can input word or two and it will show you some movies that you might be looking for.

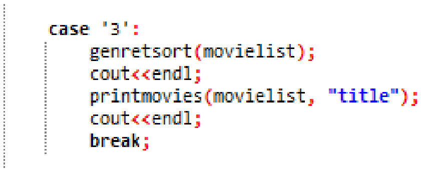


```
Select an option
1. Display all
2. Search movie
3. Sort by genre
4. Add movie
5. Remove movie
6. Exit
>> 2

Please input a title
>> Die Hard

Movie titles containing: "die hard"

Title:          Die Hard
Director:       John McTiernan
Main Actors:    Bruce Willis

Title:          Die Hard 2
Director:       Renny Harlin
Main Actors:    Bruce Willis
```

4. Sort By Genre

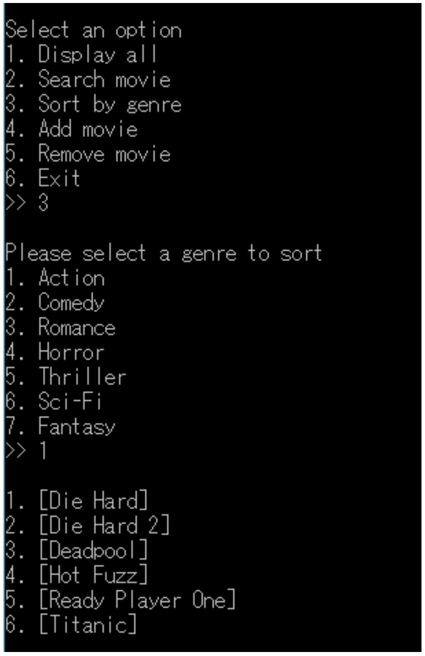


```
void genretsort(vector<Movie>& movielist)
{
    cout<<"Please select a genre to sort"<<endl;
    cout<<"1. Action"<<endl<<"2. Comedy"<<endl<<"3. Romance"<<endl<<"4. Horror"<<endl<<"5. Thriller"<<endl<<"6. Sci-Fi"<<endl<<"7. Fantasy"<<endl;
    string selection;
    cout<<">> "; getline(cin, selection);
    transform(selection.begin(), selection.end(), selection.begin(), ::tolower);

    char genre;

    if(selection == "1" || selection == "action")
        genre = 'A';

    else if(selection == "2" || selection == "comedy")
        genre = 'C';

    else if(selection == "3" || selection == "romance")
        genre = 'R';

    else if(selection == "4" || selection == "horror")
        genre = 'H';

    else if(selection == "5" || selection == "thriller")
        genre = 'T';

    else if(selection == "6" || selection == "sci-fi")
        genre = 'S';

    else if(selection == "7" || selection == "fantasy")
        genre = 'F';

    else
    {
        cout<<"Invalid genre"<<endl;
    }

    Movie current;
    int indexer;

    for(int index = 1; index < movielist.size(); index++)
    {
        current = movielist[index];
        indexer = index-1;
        while (indexer >= 0 && current.getWeight(genre) > movielist[indexer].getWeight(genre))
        {
            movielist[indexer+1] = movielist[indexer];
            indexer--;
        }
        movielist[++indexer] = current;
    }
}
```



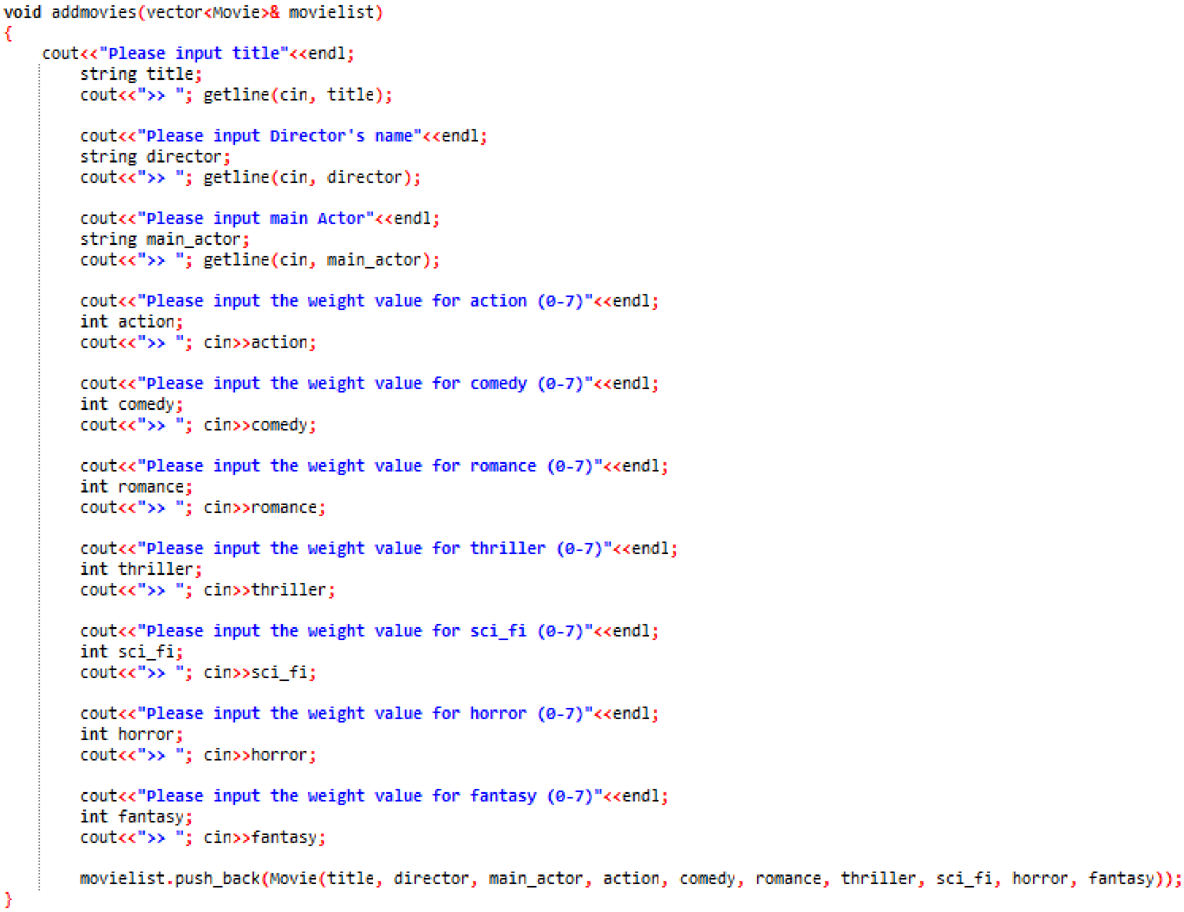
```
case '3':
    genretsort(movielist);
    cout<<endl;
    printmovies(movielist, "title");
    cout<<endl;
    break;
```

If you input 3, you can sort movies by a genre. So if you want to sort by “Action” genre, it will sort every movies that has “Action” genre from the “weightiest” to the “lightest”.



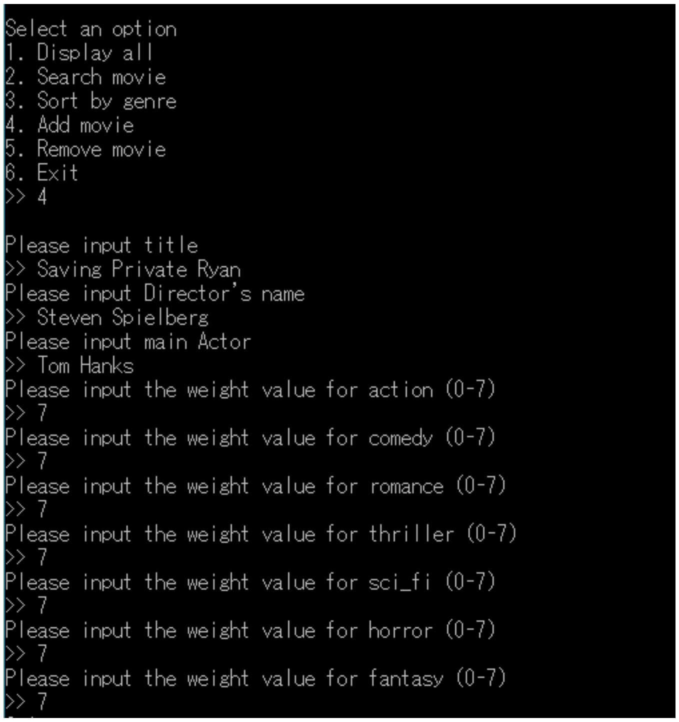
```
Select an option
1. Display all
2. Search movie
3. Sort by genre
4. Add movie
5. Remove movie
6. Exit
>> 3

Please select a genre to sort
1. Action
2. Comedy
3. Romance
4. Horror
5. Thriller
6. Sci-Fi
7. Fantasy
>> 1

1. [Die Hard]
2. [Die Hard 2]
3. [Deadpool]
4. [Hot Fuzz]
5. [Ready Player One]
6. [Titanic]
```

5. Add Movie



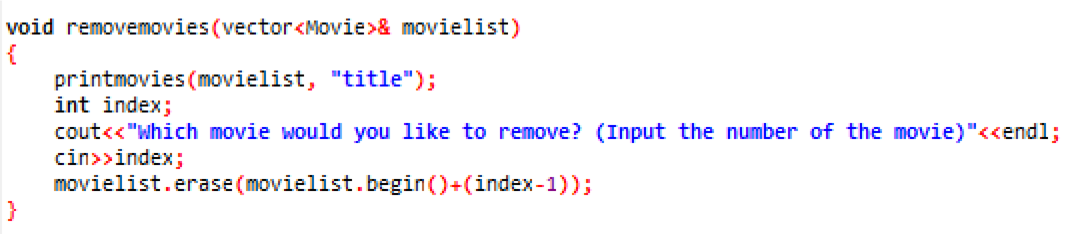
```
void addmovies(vector<Movie>& movielist)
{
    cout<<"Please input title"<<endl;
        string title;
        cout<<">> "; getline(cin, title);

        cout<<"Please input Director's name"<<endl;
        string director;
        cout<<">> "; getline(cin, director);

        cout<<"Please input main Actor"<<endl;
        string main_actor;
        cout<<">> "; getline(cin, main_actor);

        cout<<"Please input the weight value for action (0-7)"<<endl;
        int action;
        cout<<">> "; cin>>action;

        cout<<"Please input the weight value for comedy (0-7)"<<endl;
        int comedy;
        cout<<">> "; cin>>comedy;

        cout<<"Please input the weight value for romance (0-7)"<<endl;
        int romance;
        cout<<">> "; cin>>romance;

        cout<<"Please input the weight value for thriller (0-7)"<<endl;
        int thriller;
        cout<<">> "; cin>>thriller;

        cout<<"Please input the weight value for sci_fi (0-7)"<<endl;
        int sci_fi;
        cout<<">> "; cin>>sci_fi;

        cout<<"Please input the weight value for horror (0-7)"<<endl;
        int horror;
        cout<<">> "; cin>>horror;

        cout<<"Please input the weight value for fantasy (0-7)"<<endl;
        int fantasy;
        cout<<">> "; cin>>fantasy;

        movielist.push_back(Movie(title, director, main_actor, action, comedy, romance, thriller, sci_fi, horror, fantasy));
}
```

If you input 4, you can add a movie and its details. Example, I want to add “Saving Private Ryan” then I will input all the details. After that, “Saving Private Ryan” will be added to the movies data.



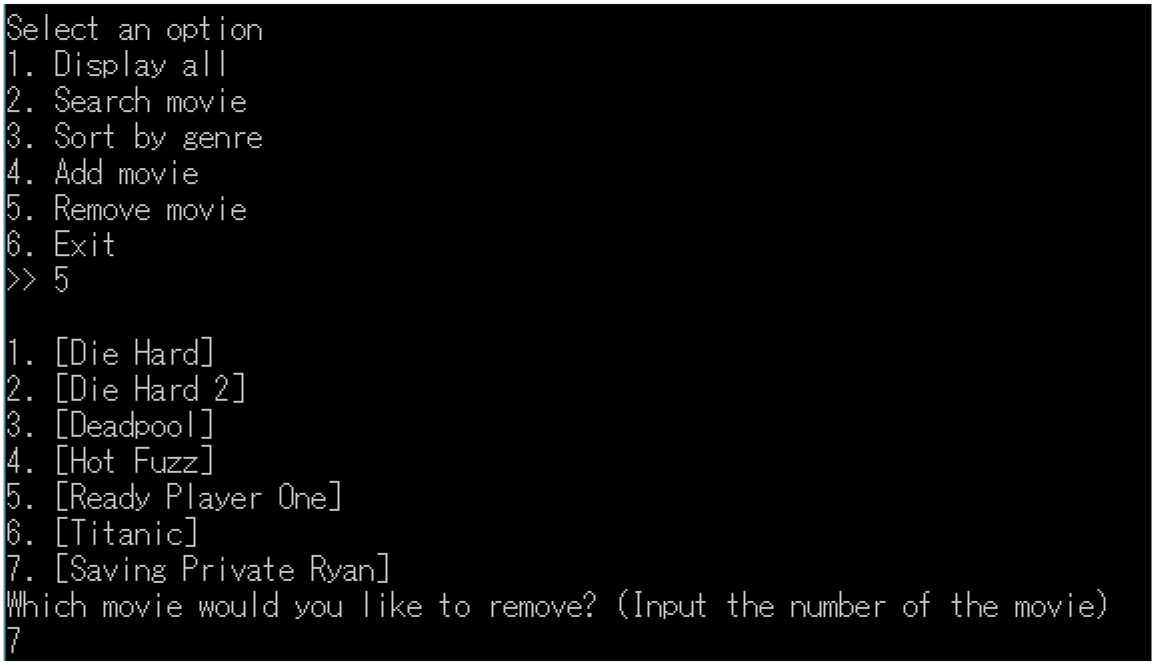
```
Select an option
1. Display all
2. Search movie
3. Sort by genre
4. Add movie
5. Remove movie
6. Exit
>> 4

Please input title
>> Saving Private Ryan
Please input Director's name
>> Steven Spielberg
Please input main Actor
>> Tom Hanks
Please input the weight value for action (0-7)
>> 7
Please input the weight value for comedy (0-7)
>> 7
Please input the weight value for romance (0-7)
>> 7
Please input the weight value for thriller (0-7)
>> 7
Please input the weight value for sci_fi (0-7)
>> 7
Please input the weight value for horror (0-7)
>> 7
Please input the weight value for fantasy (0-7)
>> 7
```

6. Remove Movie



```
void removemovies(vector<Movie>& movielist)
{
    printmovies(movielist, "title");
    int index;
    cout<<"Which movie would you like to remove? (Input the number of the movie)"<<endl;
    cin>>index;
    movielist.erase(movielist.begin()+(index-1));
}
```

If you input 5, you can remove a movie that already exist.



```
Select an option
1. Display all
2. Search movie
3. Sort by genre
4. Add movie
5. Remove movie
6. Exit
>> 5

1. [Die Hard]
2. [Die Hard 2]
3. [Deadpool]
4. [Hot Fuzz]
5. [Ready Player One]
6. [Titanic]
7. [Saving Private Ryan]
Which movie would you like to remove? (Input the number of the movie)
7
```

Links

Github: https://goo.gl/Hqa3sM

Youtube: https://goo.gl/ZB6bbN