
1.124J Foundations of Software Engineering

Problem Set 4 - Solution

Due Date: Tuesday 10/17/00

Problem 1:[40%]

sol4_1.h

// Problem Set#4 - Problem#1 solution [sol4_1.h]

#define YEARS 50
#define MONTHS 12

template<typename DataType>
int readData(DataType (&data)[YEARS][MONTHS], int &firstYear)
{
int nYears=0;
char fileName[80];

cout << "\n\n First year: " ;
cin >> firstYear ;

cout << " File with data: " ;
cin >> fileName ;
ifstream inputStreamName (fileName);

inputStreamName.clear();

while(1)
{

```

    for(int j=0 ; !inputStreamName.eof() && j<MONTHS ; j++)
        inputStreamName >> data[nYears][j];

    if(inputStreamName.eof())
break;

    nYears++;
}
cout << "\n Data for " << nYears << " years have been read\n\n";
return nYears;
}

template<typename DataType>
void writeInvertedData( DataType (&data)[YEARS][MONTHS], int firstYear,
    int lastYear, int precision)
{
    int nYears=lastYear-firstYear+1;
    char fileName[80];

    cout << " File to store data: " ;
    cin >> fileName ;
    ofstream outputStreamName(fileName);

    outputStreamName << setw(6) << "\nMonth" ;
    for(int i=0; i<nYears; i++)
        outputStreamName << setw(9) << firstYear+i ;
    outputStreamName << endl;

    for(int j=0 ; j<MONTHS ; j++)
    {
        outputStreamName.setf(ios::left);
        outputStreamName << setw(2) << " " << setw(4) << j+1 ;
        outputStreamName.unsetf(ios::left);
        outputStreamName.setf(ios::fixed);

        for(int i=0; i<nYears; i++)
        {
            outputStreamName << setprecision(precision) << setw(9) << data[i][j];
        }
        outputStreamName << endl;
    }
}

```

```
    cout << "\n Data for " << nYears << " years have been stored to file: "  
        << fileName << endl;  
}
```

sol4_1.C

```
// Problem Set#4 - Problem#1 solution [sol4_1.C]
```

```
#include <iostream.h>  
#include <iomanip.h>  
#include <stdlib.h>  
#include <fstream.h>  
#include <string>  
#include <stdio.h>
```

```
#include "sol4_1.h"
```

```
int main(void)
```

```
{  
    int firstYear, lastYear, nYears;
```

```
    // Use of the template functions for an array of ints  
    int visitors[YEARS][MONTHS];
```

```
    cout << "\n Visitors Statistics" << endl;
```

```
    nYears = readData(visitors, firstYear);  
    lastYear = firstYear + nYears - 1;
```

```
    writeInvertedData(visitors, firstYear, lastYear, 0);
```

```
    // Use of the template functions for an array of doubles  
    double income[YEARS][MONTHS];
```

```
    cout << "\n Income Statistics" << endl;
```

```
    nYears = readData(income, firstYear);  
    lastYear = firstYear + nYears - 1;
```

```
writeInvertedData(income, firstYear, lastYear, 3);
```

```
return EXIT_SUCCESS;
```

```
}
```

Problem 2:[60%]

sol4_2.h

```
// Problem Set#4 - Problem#2 solution [sol4_2.h]
```

```
#ifndef SOL_4__2H
```

```
#define SOL_4_2_H
```

```
int main(void);
```

```
int readPoints(Point ***points);
```

```
void printPoint(Point **points,int numberPoints);
```

```
void quickSortPoints(Point **p, int l, int r, double(Point::*pFun)());
```

```
int partitionPoints(Point **p, int l, int r, double(Point::*pFun)());
```

```
void releaseMemory(Point **points,int numberPoints);
```

```
#endif
```

sol4_2.C

```
// Problem Set#4 - Problem#2 solution [sol4_2.C]
```

```
#include <iostream.h>
```

```

#include <iomanip.h>
#include <stdlib.h>
#include <fstream.h>
#include <string>
#include <stdio.h>

#include "point.h"
#include "sol4_2.h"

/*****/

int main(void)
{
    Point **points;
    double (Point::*pFun)();
    int numberPoints;

    numberPoints = readPoints(&points);
    cout << "\n\t\t Points " << endl;
    printPoint(points, numberPoints);

    pFun = &Point::getX;
    quickSortPoints(points, 0, numberPoints-1, pFun);
    cout << "\n  Points sorted in X-direction" << endl;
    printPoint(points,numberPoints);

    pFun = &Point::getY;
    quickSortPoints(points, 0, numberPoints-1, pFun);
    cout << "\n  Points sorted in Y-direction" << endl;
    printPoint(points,numberPoints);

    releaseMemory(points,numberPoints);
    return EXIT_SUCCESS;
}

/*****/

/*****/

int readPoints(Point ***points)
{
    int n;
    double x,y;

```

```
cout << "Number of Points: ";
cin >> n;
```

```
*points = new Point*[n];
```

```
for(int i=0; i<n; i++)
```

```
{
```

```
    cout << "\n x = ";
```

```
    cin >> x;
```

```
    cout << "\n y = ";
```

```
    cin >> y;
```

```
    (*points)[i] = new Point(x,y);
```

```
}
```

```
return n;
```

```
}
```

```
/***/
```

```
/***/
```

```
void printPoint(Point **points, int numberPoints)
```

```
{
```

```
    for(int i=0; i<numberPoints; i++)
```

```
        cout << " Point " << i+1 << ": " << *(points[i]) << endl;
```

```
    cout << endl;
```

```
}
```

```
/***/
```

```
/***/
```

```
void quickSortPoints(Point **p, int l, int r, double(Point::*pFun)())
```

```
{
```

```
    if(l<r)
```

```
    {
```

```
        int i = partitionPoints(p, l, r, pFun);
```

```
        quickSortPoints(p, l, i, pFun);
```

```
        quickSortPoints(p, i+1, r, pFun);
```

```
    }
```

```
}
```

```
/***/
```

```

/*****/
int partitionPoints(Point **p, int l, int r, double(Point::*pFun)())
{
    Point *tmp;
    int i = l-1;
    int j = r+1;
    double value = (p[l]->*pFun)();

    while(1)
    {
        do
        j--;
        while((p[j]->*pFun)()>value);

        do
        i++;
        while((p[i]->*pFun)()<value);

        if(i<j)
        {
            tmp = p[i];
            p[i] = p[j];
            p[j] = tmp;
        }
        else
        return j;
    }
}
/*****/

```

```

/*****/
void releaseMemory(Point **points,int numberPoints)
{
    cout << "\n Releasing all the dynamically allocated"
        << " memory and exiting... " << endl<< endl;
    for(int i=0; i<numberPoints; i++)
        delete points[i];
    delete []points;
}
/*****/

```

point.h

// Problem Set#4: [point.h]

#ifndef POINT_4_H

#define POINT_4_H

class Point

{

private:

double x,y;

public:

Point(double x, double y);

double getX(void);

double getY(void);

friend ostream& operator << (ostream &i, Point &p);
};

#endif

point.C

// Problem Set#4: solution [point.C]

#include <iostream.h>

#include "point.h"

Point::Point(double x, double y)

{

this -> x = x ;

this -> y = y ;

}

double Point::getX(void)


```
{  
  return x;  
}
```

```
double Point::getY(void)  
{  
  return y;  
}
```

```
ostream& operator << (ostream &o, Point &p)  
{  
  o << " (x,y) = (" << p.x << " , " << p.y << ")" ;  
  return o;  
}
```