**10 mark questions**

Q1> Create a class Time with three data members: hours, minutes and seconds (HH:MM:SS), Overload + operator to add two time objects. Your task is to take input time , the first line contains hours, minutes and seconds (space separated), next line contains one more time hours,minutes and seconds (space separated) , your task is to add these two times and output the final answer.

Input Format:  
First line : hours , minutes and seconds (space separated)

Second line: hours , minutes and seconds (space separated)

Output Format:

Addition of two time

Solution:

#include<iostream>

using namespace std;

class Time{

int hours;

int minutes;

int seconds;

public:

Time(int h,int m,int s){

hours = h;

minutes = m;

seconds = s;

}

Time operator+(Time t){

int new\_seconds = t.seconds + seconds ;

int carry = new\_seconds/60;

new\_seconds = new\_seconds % 60;

int new\_minutes = t.minutes + minutes + carry ;

carry = new\_minutes / 60;

new\_minutes = new\_minutes % 60;

int new\_hours = t.hours + hours + carry;

new\_hours = new\_hours % 24;

return Time(new\_hours,new\_minutes,new\_seconds);

}

void display(){

cout << hours << " : " << minutes << " : " << seconds << endl;

}

};

int main(){

Time t1(11,59,59);

Time t2(12,59,59);

t1.display();

t2.display();

Time t3 = t1 + t2;

t3.display();

return 0;

}

|  |  |
| --- | --- |
| Input | Output |
| 11 59 59  12 59 59 | 0 59 58 |

Explanation:

Adding two time (11:59:59 and 12:59:59) we get 0:59:58

|  |  |
| --- | --- |
| Input | Output |
| 11 59 59  12 59 59 | 0 59 58 |
| 7 49 53  4 11 30 | 12 1 23 |
| 10 58 58  23 20 51 | 10 19 49 |
| 11 12 13  14 15 19 | 1 27 32 |
| 8 51 54  3 13 34 | 12 5 28 |

Q2>Find the maximum from the array of different data types (i.e, integer and char) using a single function using function templates. Take an input N which denotes the size of the array on the first line. The second line contains N integers, the third line contains N characters .

Display the maximum element of N integers on the first line and the maximum character from the character integer on the second line.

Input Format:  
First Line : N (denoting the size of both arrays)

Second line : N integers

Third line : N characters (space seperated)

Output Format:  
First Line: maximum from the integer array

Second Line: maximum from the character array

Constraints:

1 <= N <=100000

|  |  |
| --- | --- |
| Input | Format |
| 4  9 3 8 0  y u q f | 9  y |

Explanation:  
9 3 8 0 in sorted order is 0 3 8 9

y u q f in sorted order is f q u y

|  |  |
| --- | --- |
| Input | Output |
| 4  9 3 8 0  y u q f | 9  y |
| 12  8 3 21 14 21 10 9 9 2 7 3 1  q v n r t y s f r z r m | 21  z |
| 23  28 23 41 34 41 30 29 29 22 27 23 21 41 21 26 37 25 25 31 39 40 36 35  m z l y g f v e u l q f p d b h l q d q r r c | 41  z |
| 33  28 23 41 34 41 30 29 29 22 27 23 21 41 21 26 37 25 25 31 39 40 36 35 22 24 27 30 35 37 25 38 35 34  q f p d b h l q d q r r c r w d n x e u o q q e k l a i t g d p h | 41  x |
| 50  1130 1125 1143 1136 1143 1132 1131 1131 1124 1129 1125 1123 1143 1123 1128 1139 1127 1127 1133 1141 1142 1138 1137 1124 1126 1129 1132 1137 1139 1127 1140 1137 1136 1131 1135 1130 1124 1123 1141 1136 1137 1130 1123 1125 1128 1134 1127 1129 1139 1125  x e u o q q e k l a i t g d p h c s p i j t h b s f y f v l a d z p b f u d k k l r w q a o z m i x | 1143  z |

Q3. Define a matrix class that represents 2D array. Implement the + operator overloading to add two 2D array. Include necessary member functions and overloaded operators to perform addition. Provide a brief implementation.

Input:

First line contains number of rows ‘M’ and number of columns ‘N’

Second line contains MxN elements on first matrix

Third line contains MxN elements on second matrix

Output:

Addition of two matrix after operator overloading

Example test case:

|  |  |
| --- | --- |
| Input | Output |
| 2 3  1 2 3  2 4 5  4 5 6  8 6 7 | 5 7 9  10 10 12 |

Test cases

|  |  |
| --- | --- |
| Input | Output |
| 2 3  1 2 3  2 4 5  4 5 6  8 6 7 | 5 7 9  10 10 12 |
| 5 4  1233 1233 1232 1233  1231 1233 1233 1232  1231 1233 1231 1233  1231 1233 1232 1232  1232 1232 1231 1232  5 7 6 5  5 2 4 3  3 4 4 3  4 5 3 3  5 5 3 4 | 1238 1240 1238 1238  1236 1235 1237 1235  1234 1237 1235 1236  1235 1238 1235 1235  1237 1237 1234 1236 |
| 8 3  101 101 100  101 99 101  101 100 99  101 99 101  99 101 100  100 100 100  99 100 99  101 100 99  37 34 36  35 35 36  36 35 36  37 35 35  37 37 35  36 35 34  39 39 36  35 34 35 | 138 135 136  136 134 137  137 135 135  138 134 136  136 138 135  136 135 134  138 139 135  136 134 134 |
| 5 9  37 34 36 40 38 40 40 33 32  40 38 40 32 34 39 39 39 33  38 33 32 34 36 38 38 32 40  39 39 34 37 33 40 32 33 33  32 35 33 34 33 32 34 34 34  13 12 13 17 17 16 16 16 14  12 14 12 13 16 16 15 14 17  13 15 12 12 13 16 13 13 13  15 16 17 12 17 17 15 16 15  13 17 17 15 14 15 12 12 17 | 50 46 49 57 55 56 56 49 46  52 52 52 45 50 55 54 53 50  51 48 44 46 49 54 51 45 53  54 55 51 49 50 57 47 49 48  45 52 50 49 47 47 46 46 51 |
| 6 3  7 4 12  10 2 4  13 3 2  7 2 4  5 13 9  3 6 6  9 7 9  11 10 9  9 6 8  7 7 8  8 7 8  9 7 7 | 16 11 21  21 12 13  22 9 10  14 9 12  13 20 17  12 13 13 |

Answer:

#include<iostream>

#include<vector>

using namespace std;

class matrix {

int m;

int n;

int mat[100][100];

public:

matrix(int m1,int n1,int mat1[100][100]){

m=m1;

n=n1;

for(int i=0;i<m;i++){

for(int j=0;j<n;j++){

mat[i][j]=mat1[i][j];

}

}

}

void display(){

for(int i=0;i<m;i++){

for(int j=0;j<n;j++){

cout<<mat[i][j]<<" ";

}

cout<<endl;

}

}

matrix operator +(matrix &obj){

int ans[100][100];

for(int i=0;i<m;i++){

for(int j=0;j<n;j++){

ans[i][j] = mat[i][j] + obj.mat[i][j];

}

}

return matrix(obj.m,obj.n,ans);

}

};

int main(){

int m1,n1;

cin>>m1>>n1;

int mat1[100][100];

for(int i=0;i<m1;i++){

for(int j=0;j<n1;j++){

cin>>mat1[i][j];

}

}

int mat2[100][100];

for(int i=0;i<m1;i++){

for(int j=0;j<n1;j++){

cin>>mat2[i][j];

}

}

matrix m11(m1,n1,mat1);

matrix m2(m1,n1,mat2);

matrix m3 = m11+m2;

m3.display();

return 0;

}