Dechen Thakuri

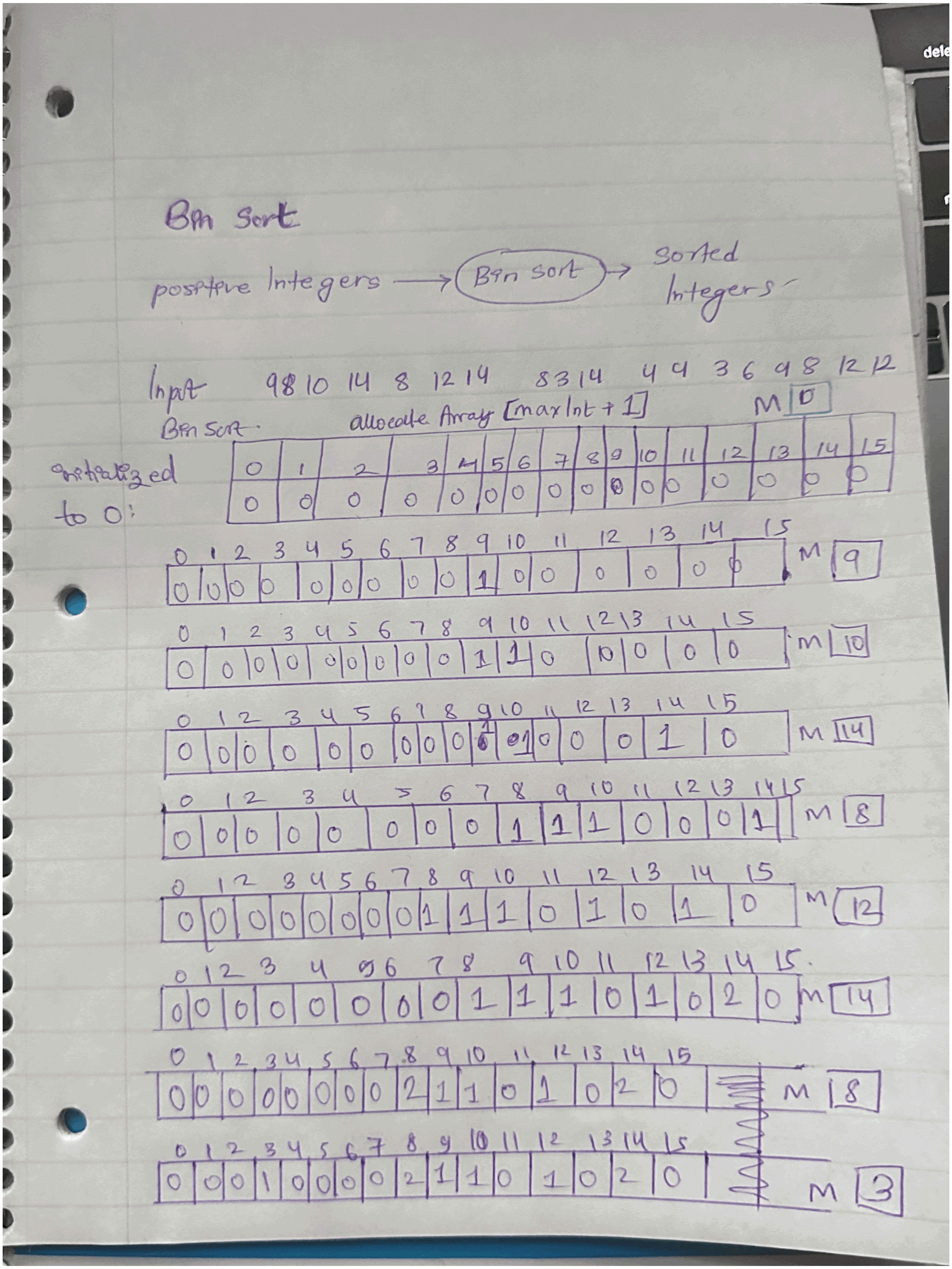
Project 3: Bin Sort(C++)

Due : 03/04/2024

```
************************************ IV. main (...) ************************************
step 0: inFileopen input file via argv [1]
outFile  open output file via argv [2] deBugFile  open deBugFile via argv [3]
step 1: maxIntfindMaxInt (inFile, deBugFile) deBugFile"In main (): maxInt =" write maxInt.
step 2: dataArydynamically allocate, size of (maxInt + 1) and initialize to zero. step 3: close
inFile
step 4: inFilere-open input file.
step 5: populateBins (inFile, dataAry, maxInt, deBugFile)
step 6: outFile"** In main (), printing non-empty bin, after populateBins () **"
printNonEmptyBins (dataAry, maxInt, outFile) //*** print to outFile!!! step 7: outFile"**In main ():
Printing sorted data. ***"
step 7: printSortedData (dataAry, maxInt, outFile)
step 8: close all files

************************************
```

Illustrations:



Bin Sort

positive Integers → Bin sort → sorted Integers

Input 9 8 10 14 8 12 14 8 3 14 4 4 3 6 9 8 12 12

Bin Sort. allocate Array [maxInt + 1] M 0

initialized to 0:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

M 9

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

M 10

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0

M 14

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	

M 8

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0

M 12

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	1	1	1	0	1	0	2	0

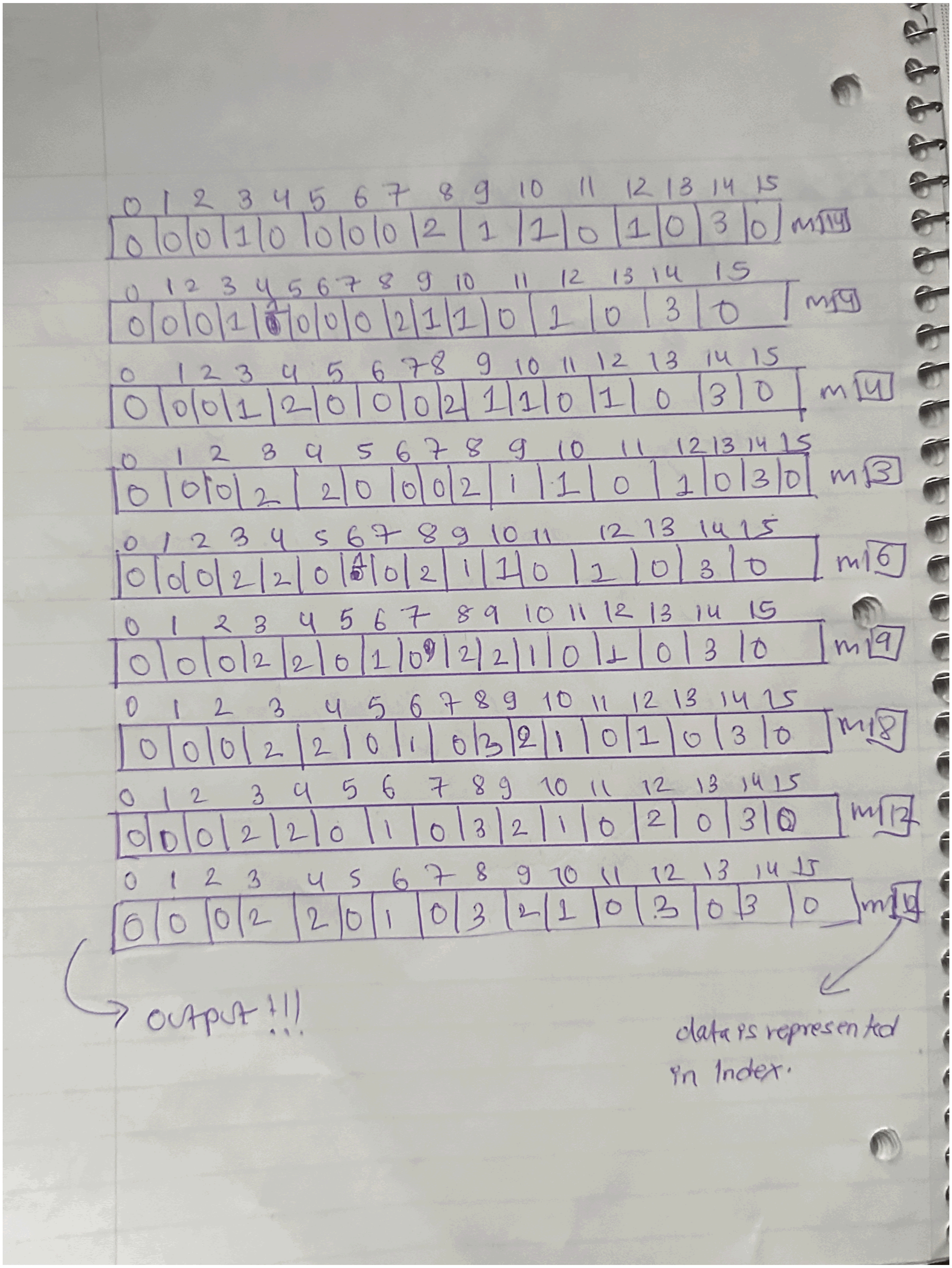
M 14

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	2	1	1	0	1	0	2	0

M 8

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	1	0	0	0	0	2	1	1	0	1	0	2	0

M 3



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	1	0	0	0	0	2	1	1	0	1	0	3	0	m 14

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	1	2	0	0	0	2	1	1	0	1	0	3	0	m 9

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	1	2	0	0	0	2	1	1	0	1	0	3	0	m 4

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	2	2	0	0	0	2	1	1	0	1	0	3	0	m 3

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	2	2	0	1	0	2	1	1	0	1	0	3	0	m 6

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	2	2	0	1	0	2	2	1	0	1	0	3	0	m 9

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	2	2	0	1	0	3	2	1	0	1	0	3	0	m 8

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	2	2	0	1	0	3	2	1	0	2	0	3	0	m 12

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	2	2	0	1	0	3	2	1	0	3	0	3	0	m 15

output !!!

data is represented in Index.

Source Code:

```
#include <iostream>
#include <fstream>


using namespace std;


class binSort {
public:
  int maxInt;
  int* dataAry;


  int findMaxInt(ifstream& inFile, ofstream& deBugFile) {
     deBugFile << "Entering FindMaxInt()" << endl;
     int maxInt;
     int data;
     bool firstnumber = true;


     while (inFile >> data) {
        if (firstnumber) {
           maxInt = data;
           firstnumber = false;
        }
        else if (data > maxInt) {
           maxInt = data;
        }
     }
     deBugFile << "In findMaxInt(): maxInt is=" << maxInt << endl;
     deBugFile << "leaving findMaxInt()" << endl;
     return maxInt;
  }


  void populateBins(ifstream& inFile, int* dataAry, int maxInt, ofstream& deBugFile) {
     deBugFile << "Entering populateBins" << endl;
     int bin;
     while (inFile >> bin) {
        if (bin < 0) {
           deBugFile << "ERROR!! In populateBins (): the data is a negative number!!!" << endl;
           exit(1);
        }
```

```
           dataAry[bin]++;
       }
       deBugFile << "In populateBins(): Printing non empty bins" << endl;
       printNonEmptyBins(dataAry, maxInt,deBugFile);
       deBugFile << "leaving populateBins()" << endl;
   }

  void printNonEmptyBins(int* dataAry, int maxInt, ofstream& deBugFile) {
      // print to debug file
      deBugFile << "Printing non-empty bins:" << endl;
      for (int i = 0; i <= maxInt; ++i) {
          if (dataAry[i] > 0) {
              deBugFile << "dataAry[" << i << "]=" << dataAry[i] << endl;
          }
      }
  }

   void printSortedData(int* dataAry, int maxInt, ofstream& outFile) {
       outFile << "In printSortedData(): Printing sorted data" << endl;
       int bin=0;
       while(bin <= maxInt) {
           while (dataAry[bin] > 0) {
               outFile << bin  << endl;
               dataAry[bin]--;
           }
           bin++;
       }
   }
};

int main(int argc, char* argv[]) {
   ifstream inFile(argv[1]);
   ofstream outFile(argv[2]);
   ofstream deBugFile(argv[3]);


   binSort sort;
   int maxInt = sort.findMaxInt(inFile, deBugFile);
   deBugFile << "In main(): maxInt = " << maxInt<< endl;


    int *dataAry = new int[maxInt+1];
   for (int i=0; i<=maxInt; ++i){
       dataAry[i]=0;
   }

   inFile.close();
   inFile.open(argv[1]);


   sort.populateBins(inFile, dataAry, maxInt, deBugFile);
   outFile <<"** In main (), printing non-empty bin, after populateBins () **";
   sort.printNonEmptyBins(dataAry, maxInt, outFile);
   outFile << "**In main (): Printing sorted data. ***";
   sort.printSortedData(dataAry, maxInt, outFile);

   inFile.close();
   outFile.close();
   deBugFile.close();

   return 0;
}
```

outFile for data1

```
** In main (), printing non-empty bin, after populateBins () **Printing non-empty bins:
dataAry[3]=2
dataAry[4]=2
dataAry[6]=1
dataAry[8]=3
dataAry[9]=2
dataAry[10]=1
dataAry[12]=3
dataAry[14]=3
**In main (): Printing sorted data. ***In printSortedData(): Printing sorted data
3
3
4
4
6
8
8
8
9
9
10
12
12
12
14
14
14
```

deBugFile for data1

```
Entering FindMaxInt()
In findMaxInt(): maxInt is=14
leaving findMaxInt()
In main(): maxInt = 14
Entering populateBins
In populateBins(): Printing non empty bins
Printing non-empty bins:
dataAry[3]=2
dataAry[4]=2
dataAry[6]=1
dataAry[8]=3
dataAry[9]=2
dataAry[10]=1
dataAry[12]=3
dataAry[14]=3
leaving populateBins()
```

outFile for data2

deBugFile for data2

```
Entering FindMaxInt()
In findMaxInt(): maxInt is=322
leaving findMaxInt()
In main(): maxInt = 322
Entering populateBins
ERROR!! In populateBins (): the data is a negative number!!!
```

outFile for data3

```
** In main (), printing non-empty bin, after populateBins () **Printing non-empty bins:
dataAry[5]=1
```

```
dataAry[8]=2
dataAry[9]=1
dataAry[10]=1
dataAry[12]=1
dataAry[13]=1
dataAry[14]=1
dataAry[16]=2
dataAry[18]=1
dataAry[22]=1
dataAry[29]=1
dataAry[31]=1
dataAry[32]=1
dataAry[36]=1
dataAry[37]=1
dataAry[55]=1
dataAry[58]=1
dataAry[66]=1
dataAry[72]=1
dataAry[77]=1
dataAry[88]=1
dataAry[91]=2
dataAry[99]=2
**In main (): Printing sorted data. ***In printSortedData(): Printing sorted data
5
8
8
9
10
12
13
14
16
16
18
22
29
31
32
36
37
55
58
66
72
```

```
77
88
91
91
99
99
```

deBugFile for data3

```
Entering FindMaxInt()
```

```
In findMaxInt(): maxInt is=99
leaving findMaxInt()
In main(): maxInt = 99
Entering populateBins
In populateBins(): Printing non empty bins
Printing non-empty bins:
dataAry[5]=1
dataAry[8]=2
dataAry[9]=1
dataAry[10]=1
dataAry[12]=1
dataAry[13]=1
dataAry[14]=1
dataAry[16]=2
dataAry[18]=1
dataAry[22]=1
dataAry[29]=1
dataAry[31]=1
dataAry[32]=1
dataAry[36]=1
dataAry[37]=1
dataAry[55]=1
dataAry[58]=1
dataAry[66]=1
dataAry[72]=1
dataAry[77]=1
dataAry[88]=1
dataAry[91]=2
dataAry[99]=2
leaving populateBins()
```