```
//*************************************************************************************/
// Example codes to transform a text file into a binary file. WuYH@ICE.CYCU
//*************************************************************************************/
#include <iostream>                     // cout, endl
#include <fstream>                      // open, is_open, close, write, ignore
#include <string>                       // string, find_last_of, substr, clear
#include <cstdlib>                      // atoi, system
#include <iomanip>                      // setw
#include <cstring>                      // strcpy

using namespace std;

#define COLUMNS       6                 // number of scores for each student
#define MAX_LEN      10                 // array size of student id and name
#define BIG_INT     255                 // integer upper bound
//*************************************************************************************/
typedef struct sT                       // type of a student record
{    char           sid[MAX_LEN];       // student id
     char           sname[MAX_LEN];     // student name
     unsigned char  score[COLUMNS];     // a set of scores in [0, 100]
     float          mean;               // the average of scores
}    studentType;
//*************************************************************************************/
int Text2Binary( );                     // get records from a text file & save as a binary file
void readBinary();
//*************************************************************************************/
int main()
{    string   fileName;

     Text2Binary(fileName);
     readBinary(fileName);
     system("pause");
     return 0;
}    // end main

int Text2Binary( )                      // get records from a text file & save as a binary file
{    fstream inFile, outFile;           // file handles
     int     stNo = 0;
```

```
do
{    cout << "Input a file number: ";
     cin >> fileName;
     if (!fileName.compare("0"))               // No file to open, so quit
          return false;
     inFile.open(("input" + fileName + ".txt").c_str(), fstream::in);      // open a file
     if (inFile.is_open())
          break;                               // successfully open a file, so leave the loop
} while(true);    //end while
fileName = "input" + fileName + ".bin";
outFile.open(fileName.c_str(), fstream::out | fstream::binary);            // create a binary file
if (outFile.is_open())
{    char      rBuf[BIG_INT];                  // buffer to keep one whole record

     while (inFile.getline(rBuf, BIG_INT, '\n'))       // retrieve one entire record from file
     {    string         temp;                // duplicate a string in another format
          studentType oneSt;                  // keep one record with all the required fields
          int            cNo = 0, pre = 0, pos = 0;    // indicators to extract each field in a record

          stNo++;                             // number of records
          temp.assign(rBuf);                  // make a duplicate of the entire record
          pos = temp.find_first_of('\t', pre);    // move to the end of the first field

          while (pos != string::npos)
          {    switch (++cNo)
               {    case 1: strcpy(oneSt.sid, temp.substr(pre, pos - pre).c_str());
                         break;               // copy a student id
                    case 2: strcpy(oneSt.sname, temp.substr(pre, pos - pre).c_str());
                         break;               // copy a student name
                    default: oneSt.score[cNo-3] = atoi(temp.substr(pre, pos - pre).c_str());
                         break;               // copy each score
               }    //end switch
               pre = ++pos;
               pos = temp.find_first_of('\t', pre);     // move to the end of the next field
          }    //end inner while
          pos = temp.find_last_of('\t');         // position of the rightmost tab
```

```
            oneSt.mean = atof(temp.substr(pos+1).c_str());        // copy the average score
```



```
            outFile.write((char *)&oneSt, sizeof(oneSt));// write the entire record into a binary file
        }    //end outer while
        outFile.close();                    // write out the binary file
    }    //end else
    inFile.close();                         // close the text file
     return stNo;                           // it succeeds if there is at least one records
}    // end Text2Binary

void readBinary(string fileName)                // get records from a file
{   fstream       binFile;                  // input file handle
    studentType oneSt;                      // keep one record with the required fields
    int             stNo = 0;               // total number of students

    binFile.open(fileName.c_str(), fstream::in | fstream::binary);    // open a binary file
    if (binFile.is_open())
    {   binFile.seekg (0, binFile.end);
        stNo = binFile.tellg() / sizeof(oneSt);
        binFile.seekg (0, binFile.beg);
        for (int i = 0; i < stNo; i++)
        {   binFile.read((char *)&oneSt, sizeof(oneSt)); // read the entire record from a binary file
            cout << "[" << i+1 << "] " << oneSt.sid << ", " << oneSt.sname << endl;
        }    //end for
    }    // end if
    binFile.close();                        // close input file
}    // end readBinary

//*********************************************************************************************/
// Keep the above codes unchanged unless its correctness can be guaranteed.
//*********************************************************************************************/
```

3