```
//fp_out2mpsx
#include <iostream.h>
#include <fstream.h>
#include <iomanip.h>
#include <conio.h>

const int RTTL=2532;
const int CTTL=2532;
// ***************************************************************************
//
//      Transform fp-out format data into MPS/X format data.
//              問題データ（ソート前）を、伊倉さん用のデータ（ＭＰＳＸフォーマッ
ト）
//              に変換するプログラム。
//
// ***************************************************************************

int ceil( int a, int b );
int floor( int a, int b );


int main( void )
{
  int            rttl, cttl;
  int            i, j, k,
                 minj;
  int            c_name[CTTL], cst[CTTL+10], h[CTTL];
  unsigned int   G[CTTL][RTTL/32+1],  wk_G[RTTL/32+1];
  char           pb_name[128],costtype[32];
  int            h1, wk_cst, wk_cname;
  unsigned int   ww;
  double         cp[CTTL], wk_cp,density;

  ifstream  fin("fp_out.txt");
  ofstream  fout("MPSdata.txt");
    if( !fin  ||  !fout )  {
      cerr << "Cannot open the file.";
      return 1;
    }

  fin  >> pb_name >> rttl >> cttl>>costtype>>density;
  cout  <<density<<endl;

  for( j = 0; j < ceil( cttl,10 ); j++)
    fin >> cst[10*j +0] >> cst[10*j +1]
        >> cst[10*j +2] >> cst[10*j +3]
        >> cst[10*j +4] >> cst[10*j +5]
        >> cst[10*j +6] >> cst[10*j +7]
        >> cst[10*j +8] >> cst[10*j +9];

  for( j = 0; j < cttl; j++)
    c_name[j] = j;

  for( j = 0; j < cttl; j++)
    for( i = 0; i < ceil( rttl,32 ); i++)
      fin >> G[j][i];
```

```
// ************************************************************
  fout << "NAME " << pb_name << endl;
  cout << "NAME " << pb_name << endl;
//
  fout << "ROWS"  << endl;
  cout << "ROWS"  << endl;
  for( i = 0; i < rttl; i++)
    fout << " G  " << 'r'
         << setw(3)<< setfill('0') << i << endl;
  fout << " N  " << "obj" << endl;
//
  fout << "COLUMNS" << endl;
  cout << "COLUMNS" << endl;
  for( j = 0; j < cttl; j++)
  {
    int count = 0;


    if( cst[j] )
    {
      fout << "    x" <<setw(4)<<setfill('0')<< j;
      fout << "     obj " << "      ";
      fout << setw(12) << setfill(' ') << cst[j] <<".";
      count++;
    }

    for( i = 0; i < rttl; i++)
    {
      ww = G[j][floor( i,32 )];
      ww = ww >> ( 31 - ( i%32 ));
      ww = ww & 1U;

      if( ww )
      {
        if( count == 0 )
          fout << "    x" <<setw(4)<<setfill('0')<< j << "  ";
                fout << "   r" << setw(3)<< setfill('0') << i << "    ";
        fout << setw(14) << setfill(' ') << "1.";

        count++;
        if( count == 2 )
        {
          fout << endl;
          count = 0;
        }
      }
    }//End of i loop.

    if( count == 1 )
      fout << endl;
  }// End of j loop.
//
  fout << "RHS" << endl;
  cout << "RHS" << endl;
```

```
  for( i = 0; i < rttl; i += 2 )
  {
    fout << "    rhs       ";

    fout << 'r' << setw(3)<< setfill('0') << i << "    ";
    fout << setw(14) << setfill(' ') << "1." << "   ";

    if( i +1 < rttl )
    {
      fout << 'r' << setw(3)<< setfill('0') << (i+1) << "    ";
      fout << setw(14) << setfill(' ') << "1.";
    }

    fout << endl;
  }

  //
  fout << "BOUNDS" << endl;
  cout << "BOUNDS" << endl;
//
  fout << "ENDATA";
  cout << "ENDATA";
// ************************************************************

  for( j = 0; j < cttl; j++)
  {
    h1 = 0;

    for( i = 0; i < rttl; i++)
    {
      ww = G[j][ floor( i,32 ) ];
      ww = ww >> ( 31 - ( i%32 ));
      ww = ww & 1U;

      if( ww )
        h1 = h1 +1;
    }

    h[j] = h1;
  }

  for( j = 0; j < cttl; j++)
  {
    if( h[j] == 0 )
    {
      fout << j << "_th column in the original input data "
                                                  "are nade of all zeros. ERROR."
<< endl;
      return 1;
    }
    else
      cp[j] = double(cst[j]) / h[j];
  }
```

```
  for( i = 0; i < rttl; i++)
  {
    for( j = 0; j < cttl; j++)
    {
      ww = G[j][floor( i,32 )];
      ww = ww >> ( 31 - ( j%32 ));
      ww = ww & 1U;

      if( ww )
        break;
    }

    if( j == cttl )
    {
      fout << i << "_th row of the original input matrix"
                << " consists of all zeros. Infeasible.¥n";
      return 1;
    }
  }

//  ---------------------- SORT  ―――――――――――――――――――
  for( j = 0; j < cttl -1; j++)
  {
    cout << setw(5) << j;

    minj  = j;

    for( k = j+1; k < cttl; k++)
      if( cp[k] < cp[minj] )
        minj  = k;

    if( minj == j )
      continue;

    wk_cst   = cst[j];
    wk_cname = c_name[j];
    wk_cp    = cp[j];
    for( i = 0; i < ceil( rttl,32 ); i++)
      wk_G[i] = G[j][i];

    cst[j]    = cst[minj];
    c_name[j] = c_name[minj];
    cp[j]     = cp[minj];
    for( i = 0; i < ceil( rttl,32 ); i++)
      G[j][i] = G[minj][i];

    cst[minj]    = wk_cst;
    c_name[minj] = wk_cname;
    cp[minj]     = wk_cp;
    for( i = 0; i < ceil( rttl,32 ); i++)
      G[minj][i] = wk_G[i];
  }
// ---------------------------------------------------------
```

```
  fin.close( );
  fout.close( );

  return 0;
}




int ceil( int a, int b )
{
  if( a % b )
    return a/b +1;

  return a/b;
}

int floor( int a, int b )
{
  return a/b;
}
```