

## fp\_out2ListR. cpp

```

//fp_out2ListR
/*
//.NET用のinclude//
#include <iostream>
#include <iomanip>
#include <fstream>

//プログラム記述の簡略化//
using namespace std;
*/

//Visual C++6.0用のinclude//
#include <iostream.h>
#include <iomanip.h>
#include <fstream.h>

//プロトタイプ宣言//
int ceil(int a, int b);
int floor(int a, int b);

// 考えられる列または行を超える数字を入れること //
/*
const int CTTL = 1090000;
const int RTTL = 5000;
*/
const int CTTL = 1100000;
const int RTTL = 5000;

char pb_name[65], unicost, csttype[17];

int rttl, ctll, ww;
int i, j, k;
unsigned int G[CTTL][(RTTL / 32) + 1];
/*
データサイズの計算
CTTL * ((RTTL / 32) + 1) * 4
RTTL = 5000, CTTL = 1090000の時、
1.09M * (5000 / 32 + 1) * 4 = 1.09M * 160 * 4 = 700MB
*/

int cst[CTTL + 10];
double rdensity;
int h[CTTL], cname[50000];
/*
cname[]の大きさは、
densityの計算で予測する。
*/

//配列はすべて0スタートで計算//
int main() {
    /*
    //ファイル読み込み.NET用//
    ifstream fin;
    fin.open("fp_out.txt");
    if(fin.fail()) {
        cerr << "Cannot open the fp_out file.";
    }
    */
}

```

fp\_out2ListR. cpp

```

        return 1;
    }
    ofstream fout;
    fout.open("ListR.txt");
    if(fout.fail()){
        cerr << "Cannot open the ListR file.";
        return 2;
    }
    */

    //ファイル読み込みVisual C++6.0用//
    ifstream fin("fp_out.txt");
    if(!fin){
        cerr << "Cannot open the fp_out file.";
        return 1;
    }
    ofstream fout("ListR.txt");
    if(!fout){
        cerr << "Cannot open the ListR file.";
        return 2;
    }

    //データ読み込み//
    fin >> pb_name >> rttl >> cttl >> csttype >> rdensity;
    cout <<"rttl = " << rttl <<" cttl = " << cttl << endl;

    //cost[]の読み込み//
    for( i = 0 ; i < ceil(cttl , 10) ; i++){
        fin >> cst[(10 * i) + 0] >> cst[(10 * i) + 1]
            >> cst[(10 * i) + 2] >> cst[(10 * i) + 3]
            >> cst[(10 * i) + 4] >> cst[(10 * i) + 5]
            >> cst[(10 * i) + 6] >> cst[(10 * i) + 7]
            >> cst[(10 * i) + 8] >> cst[(10 * i) + 9];
    }

    //G[]の読み込み//
    for( i = 0 ; i < cttl ; i++)
        for( j = 0 ; j < ceil(rttl , 32) ; j++)
            fin >> G[i][j];

    cout <<"An output working start to ListR." << endl;

    fout << rttl <<" " << cttl <<" " << endl;

    //cst[]出力//
    for( i = 0 ; i < cttl ; i++){
        fout << cst[i] <<" ";
        // 15個おきに改行//
        if(i % 15 == 14)
            fout <<endl;
    }
    //cttlが15の倍数の時は改行がない//
    if(cttl % 15)
        fout <<endl;

    //fp_outからListRへのデータ変換作業//

```

```

fp_out2ListR. cpp
for( i = 0; i < rttl; i++ ){
    h[i] = 0;
    for( j = 0; j < cttl; j++ ){
        ww = G[j][floor(i, 32)];
        ww = (ww << (i % 32));
        if(ww < 0){
            //ListRは1スタートで番号を保存しているためj+1を
            cname[h[i]] = j + 1;
            h[i]++;
        }
    }
    //列総数を出力した後改行//
    fout << h[i] <<" " << endl;
    for( k = 0 ; k < h[i] ; k++){
        fout << cname[k] <<" ";
        //15個おきに改行//
        if(k % 15 == 14)
            fout <<endl;
    }
    //ちょうどh[i]が15の倍数なら改行はいらない//
    if(h[i] % 15)
        fout << endl;
}

cout <<"An output working end to ListR." << endl;

//ファイルクローズ//
fin. close();
fout. close();
}
//プロトタイプ宣言の計算式//
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;
}

```