

```

//mpsx2fp_out
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <limits.h>
#define CTTL 2500
#define RTTL 2500
#define N 80

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

int main()
{
    char NAME[5] = "NAME", ROWS[5] = "ROWS"; //初期化子が多すぎるというエラー
    一。[4]を[5]にすることで解決
    char UG[2] = "G", UN[2] = "N", obj[32] = "obj"; //G, Nはすでに使われている
    添え字のためUG, UNに変更 (U=Uni)
    char COLUMNS[8] = "COLUMNS", RHS[4] = "RHS", rhs[4] = "rhs";
    char BOUNDS[7] = "BOUNDS", ENDATA[7] = "ENDATA";
    char s[N], s1[N], s2[N], s3[N], s4[N];
    int rttl, cttl, cst[CTTL+10], clines, cfiller, erno;
    char pb_name[64], csttype[16];
    double edensity;
    unsigned int G[CTTL][RTTL/32+1], ww, UU, U1;
    int i, j, k;

    //page 5/11

    erno = 0;
    rttl = cttl = 0;
    i = j = k = 0;
    U1 = 1;

    FILE *mpsx, *fout;
    mpsx = fopen("MPS.txt", "r");
    fout = fopen("fp_out.txt", "w");

    if(!mpsx || !fout)
    {
        printf("Cannot open the files.¥n");
        return 10;
    }

    //page 6/11

    fscanf(mpsx, "%s %s", s1, pb_name);
    printf("%s %s¥n", s1, pb_name);

    if(strcmp(s1, NAME))
    {
        erno = 20;
        printf("Error!! Num is %d¥n", erno);
        return erno;
    }
}

```

```

fscanf(mpsx, "%s", s1);
printf("%s\n", s1);

if(strcmp(s1, ROWS))
{
    errno = 30;
    printf("Error!! Num is %d\n", errno);
    return errno;
}

//rttl = 0; //すでに初期化は実行済み
while(1) //無限ループをcontinueやbreakで制御
{
    fscanf(mpsx, "%s %s", s1, s2);
    printf("%s %s\n", s1, s2);
    if(!strcmp(s1, UG)) //UG=Gに変更していることに注意
        rttl++;
    else
        break;
}
printf("rttl = %d\n", rttl); //debug

if(strcmp(s1, UN)) //UN=Nに変更していることに注意
{
    errno = 40;
    printf("Error!! Num is %d\n", errno);
    return errno;
}

strcpy(obj, s2);
printf("%s\n", obj); //debug

fscanf(mpsx, "%s", s1);
printf("%s\n", s1);

if(strcmp(s1, COLUMNS))
{
    errno = 50;
    printf("Error!! Num is %d\n", errno);
    return errno;
}

//Gのゼロクリア
for(j = 0; j < CTTL; j++)
    for(i = 0; i < ceil(RTTL, 32); i++)
        G[j][i] = 0;

//page 7/11

/*
strncmp(char *a, char *b, int n);
aが指す文字列の配列とbが指す文字列の配列のn番目までの文字列に対して、
比較を行う関数
*/
//cttl = 0; //すでに初期化は実行済み
i = j = 0;

```

mpsx2fp_out.cpp

```
while(1)//無限ループをcontinueやbreakで制御
{
    fscanf(mpsx,"%s",s);
    printf("%s",s);
    if(!strcmp(s,RHS))
    {
        break;
    }
    else if(!strncmp(s,"x",1))//これがs[0]と文字列xとの比較の書き方
    {
        printf("debug %s¥n",s);
        j = atoi(s+1);
        continue;
    }
    else if(!strncmp(s,"r",1))
    {
        i = atoi(s+1);
        fscanf(mpsx,"%s",s1);
        printf("%s",s1);
        UU = (U1 << (31-i%32));
        ww = G[j][i/32];
        ww = (ww | UU);
        G[j][i/32] = ww;
        k++;
        continue;
    }
    else if(!strcmp(s,obj))
    {
        fscanf(mpsx,"%s",s1);
        printf("%s",s1);
        cst[j] = atof(s1);
        ct1++;
        continue;
    }
    else
    {
        errno = 55;
        printf("Error!! Num is %d¥n i = %d j = %d¥n",errno,i,j);
        return errno;
    }
}
//continue必要なかったかも

//page 8/11
if(ct1-1 != j)
{
    errno = 60;
    printf("Error!! Num is %d¥n i rttl j ct1¥n%d %d %d
%d¥n",errno,i,rttl,j,ct1);
    return errno;
}

//RHSチャプタの処理1
for(i = 0;i < rttl/2-1;i++)
{
    fscanf(mpsx,"%s %s %s %s %s",s,s1,s2,s3,s4);
```

```

                                mpsx2fp_out.cpp
    printf("%s %s %s %s %s\n", s, s1, s2, s3, s4);
}

if(rttil % 2 == 0)
{
    fscanf(mpsx, "%s %s %s %s %s", s, s1, s2, s3, s4);
    printf("%s %s %s %s %s\n", s, s1, s2, s3, s4);
}
else
{
    fscanf(mpsx, "%s %s %s", s, s1, s2);
    printf("%s %s %s\n", s, s1, s2);
}

fscanf(mpsx, "%s", s);
printf("%s\n", s);
if(strcmp(s, BOUNDS))
{
    errno = 80;
    printf("Error!! Num is %d\n", errno);
    return errno;
}

fscanf(mpsx, "%s", s);
printf("%s\n", s);
if(strcmp(s, ENDATA))
{
    errno = 90;
    printf("Error!! Num is %d\n", errno);
    return errno;
}

//page 9/11
printf("Successful end of reading an MPSX data.\n");

printf("Input csttype (uni_cost or random_cost) ");
scanf("%s", &csttype);

if(rttil % 32 != 0)
{
    ww = UINT_MAX >> (rttil % 32); //UNIT_MAXはU1と同じ
    //G未使用部分に1をフィル
    for(j = 0; j < cttil; j++)
    {
        G[j][rttil/32] = (G[j][rttil/32] | ww);
    }
}

edensity = float(k) / (float(rttil) * float(cttil));
fprintf(fout, "%s %d %d %s %f\n", pb_name, rttil, cttil, csttype, edensity);
printf("%s %d %d %s %f\n", pb_name, rttil, cttil, csttype, edensity);
clines = ceil(cttil, 10);
cfiller = 10 - (cttil % 10);

//fill -1 for cst[]
if(cttil % 10) //フローチャートの内容が怪しかったので過去のプログラムから

```

引用

```

                                mpsx2fp_out.cpp
for (j = 0; j < cfiller; j++)
    cst[cttl + j] = -1;

//Output cst[] into fout
for (i = 0; i < clines; i++)
{
    fprintf(fout, "%d %d %d %d %d %d %d %d %d %d %d\n"
            , cst[10*i], 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]);
    printf("%d %d %d %d %d %d %d %d %d %d %d\n"
            , cst[10*i], 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の出力制御子
for (j = 0; j < cttl; j++)
{
    for (i = 0; i < ceil(rttl, 32); i++)
    {
        fprintf(fout, "%lu\n", G[j][i]); //unsigned int の出力は%lu
とする。%dではないことに注意！
        printf("%lu\n", G[j][i]);
    }
}
fclose(mpsx);
fclose(fout);
}
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;
}

```