

24 先週の quiz

—— 適宜省略した答でよい ——

```
int n[XMAX][YMAX],nextn[XMAX][YMAX];
int x,y,dx,dy;
int nep;      /* まわり 8 か所中の ON の個数 */

/* ここで n[x][y] の初期化 */

for(x=1; x<XMAX-1; x++){
  for(y=1; y<YMAX-1; y++){
    nep=0;
    for(dx=-1; dx<=1; dx++){
      for(dy=-1; dy<=1; dy++){ /* 注目点 (x,y) から+1,0,-1の9個所 */
        if( ( dx!=0 || dy!=0 ) /* 注目点 (x,y) 自身は除く*/
            && n[x+dx][y+dy]==ON ){
          nep++;
        }
      }
    }
  }

  /* 次の時刻の状態は、いったん nextn[x][y] に保存する。
     いきなり n[x][y] にいれると、他の点の nep が変わっちゃってまずい */
  if( nep >= 4){
    nextn[x][y]=ON;
  } else {
    nextn[x][y]=OFF;
  }
}

for(x=1; x<XMAX-1; x++){
  for(y=1; y<YMAX-1; y++){
    n[x][y]=nextn[x][y];
  }
}
```

¹<http://sparrow.math.ryukoku.ac.jp/~hig/compsci/>

²<mailto:hig@math.ryukoku.ac.jp>, <http://www.math.ryukoku.ac.jp/~hig/>,
へや 1-508, でんわ 077-543-7501

25 今週の quiz

プログラムが,

- `int main(int argc, char **argv)` の定義を含む `prog.c`
- `double g_u_r(void)` の定義を含む `random.c`
- `double g_u_r(void)` の関数プロトタイプ宣言を含む `random.h`
- `int get_matint(int *m)` の定義を含む `matrix.c`
- `int get_matint(int *m)` の関数プロトタイプ宣言を含む `matrix.h`

に分割されており,

— prog.c —

```
#include "random.h"
#include "matrix.h"

int main(int argc, char **argv){
/*略*/
}
```

— random.c —

```
#include "random.h"

double g_u_r(void){
/*略*/
}
```

— matrix.c —

```
#include "matrix.h"

int get_matint(int *matrix){
/*略*/
}
```

のようになっている。

分割コンパイルのための Makefile をかこう。

単一ファイルの場合

```
prog.c
#include <stdio.h>
#include <math.h>

/*関数プロトタイプ宣言*/
double g_u_r(void);

int main(int argc, char **argv){
    double r,s;
    r=g_u_r();
    s=sin(1.2);
    printf("%f %f\n", r,s);
    return 0;
}

/*関数の定義*/
double g_u_r(void){
    /* 略 */
    return some_double;
}
```

まとめてコンパイル

```
$ cc -o prog prog.c -lm
```

分割コンパイル

```
$ cc -c prog.c
# prog.o ができる

$ cc -o prog prog.o -lm
# prog ができる
```

Makefile

```
prog: prog.o
    cc -o prog prog.o -lm

prog.o: prog.c
    cc -c prog.c
```

注: Makefile の行頭の空白は, スペース複数個でなく Tab 1 個.

複数ファイルの場合

```
prog.c
#include <stdio.h>
#include <math.h>
#include "random.h" /* ! */

int main(int argc, char **argv){
    double r,s;
    r=g_u_r();
    s=sin(1.2);
    printf("%f %f\n", r,s);
    return 0;
}
```

random.h

```
double g_u_r(void);
```

random.c

```
#include "random.h"

double g_u_r(void){
    /* 略 */
    return some_double;
}
```

まとめてコンパイル

```
$ cc -o prog prog.c random.c -lm
```

分割コンパイル

```
$ cc -c prog.c
$ cc -c random.c
$ cc -o prog prog.o random.o -lm
```

Makefile

```
prog: prog.o random.o
    cc -o prog prog.o random.o -lm

prog.o: prog.c random.h
    cc -c prog.c

random.o: random.c random.h
    cc -c random.c
```