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

#define L 200
#define NUM_ITER 100

typedef unsigned long long RANDOM_TYPE;
typedef char SPIN;

#define MYRAND64  myrand64 = (6364136223846793005UL * myrand64)
#define MASSIMO_RAND64      0XFFFFFFFFFFFFFFFFUL
RANDOM_TYPE myrand64;
double inv_massimo_rand64 = 1.0L / (double)MASSIMO_RAND64;

SPIN C[L], CP[L];
int regola=90;

#define FAILURE -1
#define SUCCESS 1

/*************************************************************************
**/
int main(void)
{
  long time, site;


    for(site=0; site<L; site++){
      C[site]=0;
      CP[site]=0;
    }
  C[L/2] = 1;
```

```
  printf("# la regola e' %d\n",regola);

  for(time=0; time<NUM_ITER; time++){
    for(site=0; site<L; site++){
      printf("X %ld T %ld CELLA %d\n",site,time,C[site]);
    }
    /* l'update e' sincrono */
    for(site=1; site<L-1; site++){
      if(regola==90){
        /* regola 90 che e' lo XOR dei vicini */
        CP[site] = C[site-1]^C[site+1];
      } else if(regola == 150){
        /* regola 150 che e' lo XOR dei vicini e lo XOR con se stesso*/
        CP[site] = (C[site-1]^C[site+1])^C[site];
      } else{
        printf("interruzione programma, regola non definita\n");
        exit(-9);
      }
    }
    /* rimetto CP in C */
    for(site=0; site<L; site++){
      C[site] = CP[site];
    }
  }
  for(site=0; site<L; site++){
    printf("X %ld T %ld CELLA %d\n",site,time,C[site]);
  }
  return SUCCESS;
}
```