```c
typedef struct
{
  int n;
  int* pTrouve;
  int debut;
  int fin;
} data;

int est_premier(int n)
{
  data d[NB_THREADS];
  pthread_t T[NB_THREADS];
  int trouve = 0;
  int i;
  int rc = racine_carre(n);
  for (i = 0; i < NB_THREADS; ++i)
  {
    d[i].n = n;
    d[i].pTrouve = &trouve;
    d[i].debut = 2 + i*(rc-2)/NB_THREAD;
    d[i].fin = 1 + (i+1)*(rc-2)/NB_THREAD;
    pthread_create(&T[i],NULL,(void *(*)(void*))div_dans_inter,(void*)&d[i]);
  }
  for (i=0; i < NB_THREADS; ++i)
  {
    pthread_join(T[i],NULL);
  }
  return !trouve;
}

void div_dans_inter(data *pd)
{
  int i;
  for (i = pd -> debut; i <= pd -> fin &&(!*pd->pTrouve) ; ++i) {
    if (pd -> n % i == 0) { *pd -> pTrouve = 1; }
  }
}
}
```