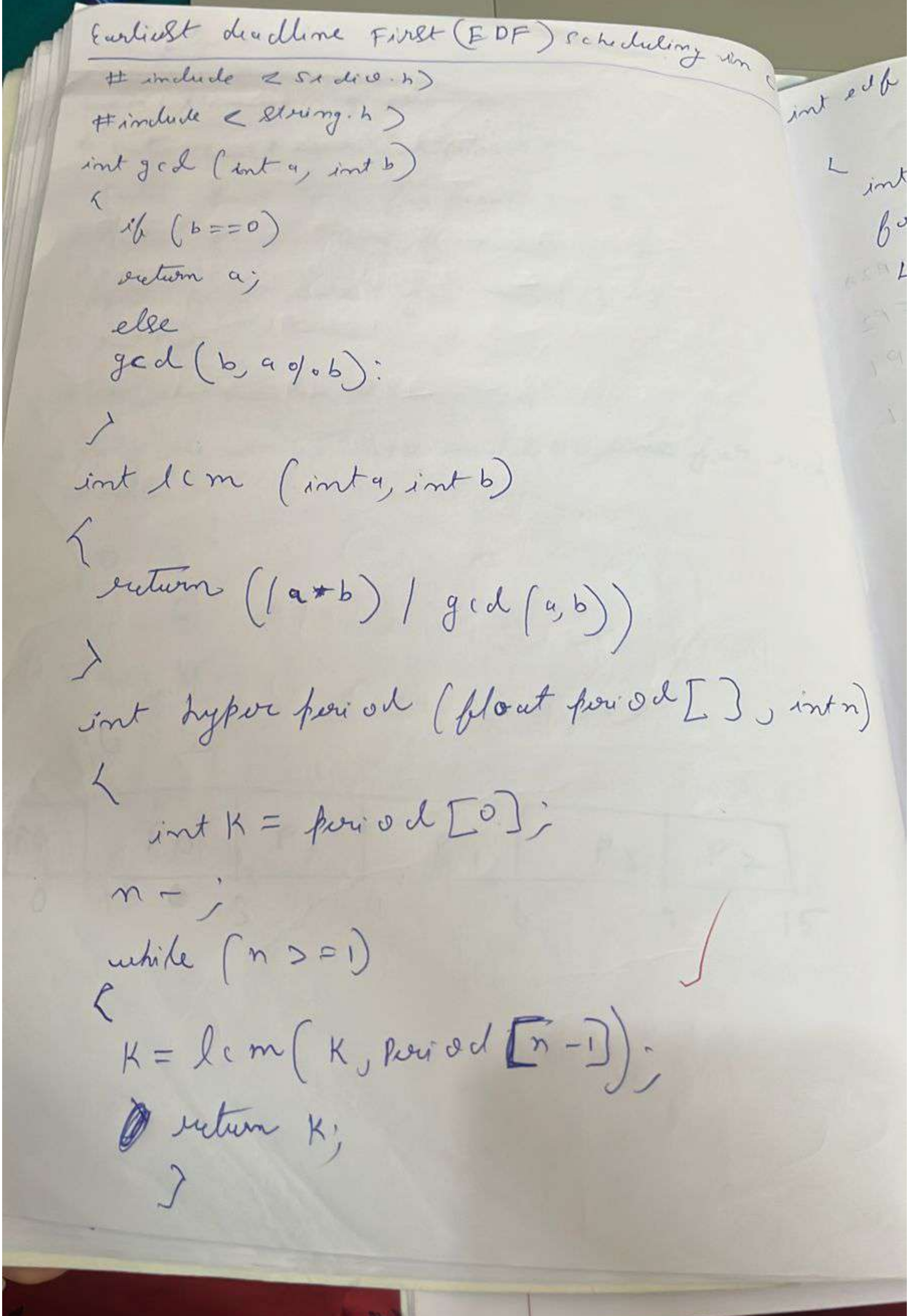

Earliest deadline First (EDF) scheduling in

```
# include <stdio.h)
#include <string.h)
int gcd (int a, int b)
{
  if (b==0)
  return a;
  else
  gcd(b, a%b):
}
int lcm (int a, int b)
{
 return ((a*b) / gcd(a,b))
}
int hyper period (float period[], int n)
{
    int K = period[0];
  n - ;
  while (n>=1)
 {
  K = lcm(K, Period[n-1]);
  return K;
   }
```

int edf

{

int

fo

```
                                                    the burst time for the process
int edf ( float * period, int n, int t, float *
          ~~dead~~  ~~dead~~                                        deadline)
{
 int i, small = 1000.0f, small index = 0;
 for ( int i = 0; i < n; i++)
  {
   if ( period[i] < small && ( period[i] - t) <=
                                                   deadline
                                                     [i])
  {
  small = period[i];
  small index = i;
      }
    }
 if ( small = = 1000.0f )
 { return -1;
 return small index;
  }
int main ()
{
 int i, n, c, d, k, j, next time = 0, time = 0,
       task, preemption_count;
 float exec [20], period [20], individual_util
                                         [20],
                                    flag [20],
                       release [20],

 ~~deadline~~
```

deadline

output

```
Enter the number of processes : 3
Enter execution times : 1 5 7
Enter deadlines : 2 4 7

0 P1 P 2 2 P 1 3 P 2 4 P1 5 P 2 6 P 17 P2
P1 9 P2 10 P1 11 P3 12 P1 13 P2 14 15 P2
16 P1 17 P2 18 P1 19 P2 20 P1 R1 P2 P1
23 P3 24 P1 25 P2 26 P1 27 PR 28 P1
```