Write a C progam to schedule Rate monotonic Earliest deadline first procedure

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

int et[10], i, n, dl[10], p[10], ready[10], flag = 1;
int lcm(int a, int b){
    int max = (a>b)? a: b;
    while (1){
        if (max % a == 0 && max % b == 0)
            return max;
        max++;
    }
}

int lcmarray(int arr[], int n){
    int result = arr[0];
    for(int i=1; i<n; i++){
        result = lcm(result, arr[i]);
    }
    return result;
}

void main
int time = lcmfrom(dl, n);
int op=0, pr=0, pre = pr;

while(op<time){
    for(i=0; i<n; i++){
        if (op%dl[i]==0){
            ready[i]=1
    }
}
```

```
flag=0;
for(i=0;i<n;i++){
    if(ready[i]==1){
        flag=1;
        break;
    }
}
if(flag==0){
    pr=-1;
}else{
    pr=-1;
    for(i=0;i<n;i++){
        if(ready[i]==1){
            if(pr==-1||dl[i]<dl[pr]){
                pr=i;
            }
        }
    }
}
if(pr!=pre){
    if(pr==-1){
        printf("%d Idle",op);
    }else{
        printf("%d p-%d ",op,pr+1);
    }
}
op++;
if(pr!=-1){
    p[pr]=p[pr]-1;
    if(p[pr]==0){
        pr[pr]=dl[pr];
        ready[pr]=0
}}
```

```
    pre=pr;
}
printf("\n");
}
void edf(){
int time=sum(dl
int op=0,pr=0,pre=
and flag,i;
while(op<=time){
    for(i=0;i<n;i
        if(op==dl[i]
            ready[i
    }
}
flag=0;
for(i=0;i<n;i++
    if(ready[i]==1
        flag=1;
        break;
}}
if(flag==0){
    pr=-1;
}else{
    pr=-1;
    for(i=0;i<n;
        if(ready[
            if(pr
}}}}
```

```
        pre = pr;
    }
    printf("\n");
}
void edf(){
int time = lcmArray(dl, n);
int op=0, pr=0, pre=-1;
int flag, i;

while(op<time){
    for(i=0; i<n; i++){
        if(op%dl[i]==0){
            ready[i] = 1;
        }
    }
    flag = 0;
for(i=0; i<n; i++){
    if(ready[i]==1){
        flag = 1;
        break;
}
}

if(flag==0){
    pr=-1;
}else{
    pr=-1;
    for(i=0; i<n; i++){
        if(ready[i]==1){
            if(pr==-1 || p[i]<p[pr]){
                pr=i;
}}}}}
```

```
if(pr != pre){
  if(pr == -1){
  printf("\t\d Idle ", op);
  }else{
    printf("\t\d P\d ", op, pr+1);
  }}
  op++;
  if(pr != -1){
  p[pr] = p[pr]-1;
    if(p[pr]==0){
      p[pr] = ct[pr];
      ready[pr] = 0;
  }}
    pre = pr;
  } printf("\n");
}

int main(){
  int ch, k=1;
  while(k){
  printf("Enter your choice \n1. Round robin \n2. Priority
        \n3. Exit\n");
  scanf("%d", &ch);
  if(ch==3)
  exit(0);
  printf("Enter no of process: ");
  scanf("%d", &n);
  printf("Enter Execution time\n");
  for(i=0; i<n; i++)
      scanf("%d", &ct[i]);
```

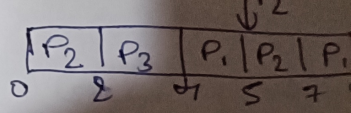
```
for(i=0; i<n; i++)
    p[i] = ct[i];
for(i=0; i<n; i++)
    ready[i] = 0;
switch(ch){
  case 1: rr();
          break;
  case 2: sjf();
          break;
  default: printf("Bu
  }
}
return 0;
}
```

Output:

Enter the no of proces

Enter execution time

Enter deadline



```
for (i=0; i<n; i++)
    r[i] = et[i];
for (i=0; i<n; i++)
    ready[i] = 0;
switch(ch){
    case 1: rms();
            break;
    case 2: edf();
            break;
    default: printf("Invalid choice\n");
    }
}
return 0;
}
```

$U_i \leq n(2^{1/n} - 1)$

Output:

Enter the no of process: 3

Enter execution times: 3 2 2

Enter deadlines 8 5 10

14/7/23