

2) Earliest-Deadline First

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

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

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

```
void sort(int proc[], int d[], int b[], int p[],  
int n)
```

```
{
```

```
int temp=0;
```

```
for(int i=0; i<n; i++)
```

```
{
```

```
for(int j=i; j<n; j++)
```

```
{  
if (d[j] < d[i])
```

```
temp = d[j];
```

```
d[j] = d[i];
```

```
d[i] = temp;
```

```
temp = p[i];
```

```
p[i] = p[j];
```

```
p[j] = temp;
```

```
temp = b[j];
```

```
b[j] = b[i];
```

```
b[i] = temp;
```

```
temp = proc[i];
```

```
proc[i] = proc[j];
```

```
proc[j] = temp;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
int gcd(int a, int b)
{
```

```
    int r;
    while(b > 0)
    {
        r = a % b;
        a = b;
        b = r;
    }
```

```
    return a;
}
```

```
int lcmul(int p[], int n)
{
```

```
    int lcm = p[0];
    for(int i = 1; i < n; i++)
```

```
    {
        lcm = (lcm * p[i]) / gcd(lcm, p[i]);
    }
```

```
    return lcm;
}
```

```
void main()
```

```
{
```

```
    int n;
```

```
    printf("Enter the no. of processes:");
```

```
    scanf("%d", &n);
```

```
    int pro[n], b[n], p[n], d[n], rem[n];
```

```
    printf("Enter the CPU burst times:");
```

```
    for(int i = 0; i < n; i++)
```

```
    {
        scanf("%d", &b[i]);
```

```
    }
    rem[i] = b[i];
```



```

int minDeadline = 1e9;
int taskToExecute = -1;
for (int i = 0; i < n; i++)
{
    if (rem[i] > 0 && nextDeadline[i] < minDeadline)
    {
        minDeadline = nextDeadline[i];
        taskToExecute = i;
    }
}

if (taskToExecute != -1)
{
    printf("%d ms: Task %d is running. Wn, time,
           proc(taskToExecute));
    rem[taskToExecute]--;
}
else
{
    printf("%d ms: CPU is idle. nn, time);
}
time++;
}
}

```

Output:-

Enter the number of processes: 3
Enter the CPU burst times:

3 2 2

Enter the deadlines:

7 4 8

Enter the time periods:

20 5 10

Earliest Deadline Scheduling:

PZD	Burst	Deadline	period
2	2	4	5
1	2	7	20
3	3	8	10

Scheduling occurs for 20 ms

0ms: Task 2 is running

1ms: Task 2 is running

2ms: Task 1 is running

3ms: Task 1 is running

4ms: Task 1 is running

5ms: Task 3 is running

6ms: Task 3 is running

7ms: Task 2 is running

8ms: Task 2 is running

9ms: CPU is idle

10ms: Task 2 is running

11ms: Task 2 is running

12ms: Task 3 is running

13ms: Task 3 is running

14ms: CPU is idle

15ms: Task 2 is running

16ms: Task 2 is running

17ms: CPU is idle

18ms: CPU is idle

19ms: CPU is idle