### Write a C program to simulate Earliest deadline first scheduling.

**PROGRAM:**

#include <stdio.h>

#define MAX 10

typedef struct {

int pid;

int burst;

int deadline;

int period;

int remaining\_burst;

int next\_deadline;

} Process;

int main() {

int n;

Process p[MAX];

printf("Enter the number of processes: ");

scanf("%d", &n);

printf("Enter the CPU burst times:\n");

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

scanf("%d", &p[i].burst);

p[i].remaining\_burst = p[i].burst;

p[i].pid = i + 1;

}

printf("Enter the deadlines:\n");

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

scanf("%d", &p[i].deadline);

p[i].next\_deadline = p[i].deadline;

}

printf("Enter the time periods:\n");

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

scanf("%d", &p[i].period);

}

printf("\nEarliest Deadline Scheduling:\n");

printf("PID\tBurst\tDeadline\tPeriod\n");

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

printf("%d\t%d\t%d\t\t%d\n", p[i].pid, p[i].burst, p[i].deadline, p[i].period);

}

int time = 0, total\_time = 20;

printf("\nScheduling occurs for %d ms\n\n", total\_time);

while (time < total\_time) {

int earliest = -1;

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

if (p[i].remaining\_burst > 0) {

if (earliest == -1 || p[i].next\_deadline < p[earliest].next\_deadline) {

earliest = i;

}

}

}

if (earliest != -1) {

printf("%dms : Task %d is running.\n", time, p[earliest].pid);

p[earliest].remaining\_burst--;

} else {

printf("%dms : CPU is idle.\n", time);

}

time++;

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

if (time % p[i].period == 0) {

p[i].remaining\_burst = p[i].burst;

p[i].next\_deadline = time + p[i].deadline;

}

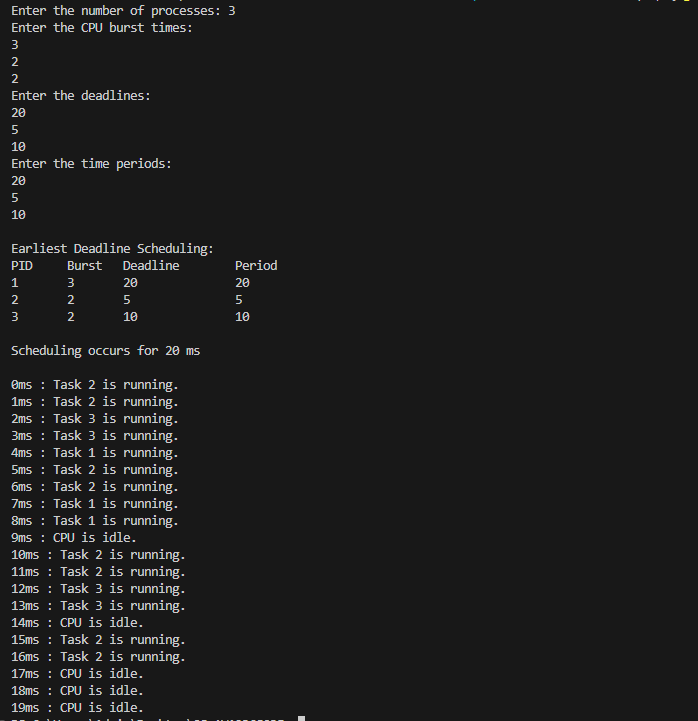
}

}

return 0;

}

**OUTPUT:**

****