

۱- (نظری) با استفاده از الگوریتم EDF می‌خواهیم مجموعه وظایف زیر را بر روی یک پردازنده زمان‌بندی کنیم. زمان‌بندپذیری این مجموعه وظایف را با استفاده از رویکرد Reducing Test Intervals بررسی کنید.

#	وظیفه	زمان اجرا	مهلت زمانی	دوره
1	τ_1	2	5	6
2	τ_2	$\lceil a / 2 \rceil$	$8 + \lceil a / 2 \rceil$	12
3	τ_3	2	9	9

به علامت براکت‌ها و جهت آنها توجه کنید.

با استفاده از رقم آخر شماره دانشجویی خود، جدول بالا را تکمیل نمایید. به جای رقم ۰، ۵ قرار دهید.

۲ - نظری) با سه الگوریتم EDF، DM و RM مجموعه وظایف زیر را بر روی یک پردازنده زمان‌بندی کنید.

#	وظیفه	زمان اجرا	مهلت زمانی	دوره
1	τ_1	$\lfloor a / 2 \rfloor$	3a	3a + 1
2	τ_2	$\lfloor b / 2 \rfloor$	3b	3b + 3
3	τ_3	c	3c	3c
4	τ_4	d	3d	3d + 2

با استفاده از ارقام شماره دانشجویی خود، جدول بالا را تکمیل نمایید. به این صورت که چهار رقم آخر شماره دانشجویی شما مشخص کننده \overline{abcd} است. به جای رقم ۰، ۵ قرار دهید. به عنوان مثال شماره دانشجویی ۴۰۰۱۹۰۲۳۴ به صورت $\overline{abcd} = 5234$ خواهد بود. خروجی سه نمودار زمان‌بندی (تا لحظه ۴۰) برای سه رویکرد EDF، RM و DM است. توجه کنید که هرکجا که مهلت زمانی وظیفه‌ای از دست رفت، زمان‌بندی را متوقف کنید.

۳ - پیاده‌سازی) در یک سامانه بی‌درنگ تعدادی وظیفه پریودیک داریم. برنامه‌ای بنویسید که زمان‌بندپذیری این وظایف را تحت دو الگوریتم DM و RM بررسی کنید.

- توجه کنید که فقط باید زمان‌بندپذیری وظایف را بررسی کنید.
- هر زبان برنامه نویسی‌ای می‌توانید استفاده کنید.
- ورودی برنامه شما یک فایل Json به صورت زیر است:

```
[
  {
    "taskId": 1,
    "executionTime": 5,
    "period": 10,
    "deadline": 10
  },
  {
    "taskId": 2,
    "executionTime": 6,
    "period": 20,
    "deadline": 18
  },
  {
    "taskId": 3,
    "executionTime": 1,
    "period": 5,
    "deadline": 5
  }
]
```

- خروجی شما نیز یک فایل Json مشابه زیر باید باشد.

```
{
  "rm": false,
  "dm": false
}
```