

توضیح مسئله‌ی Dining Philosophers

تهیه و تنظیم: مبین خیبری

شماره دانشجویی: 994421017

استاد راهنما: دکتر لیلا شریفی

توضیحات عمومی:

مسئله‌ی Dining Philosophers تعدادی فرضی اولیه دارد که در زیر به آن‌ها اشاره کرده‌ایم:

- یک یا چند فیلسوف دور یک میز دایره‌ای نشسته‌اند.
- یک ظرف بزرگ حاوی اسپاگتی در وسط میز قرار دارد.
- برای خوردن اسپاگتی، هر نفر به 2 چنگال نیاز دارد.
- تنها به تعداد فیلسوفان چنگال در اختیار داریم.
- هر فیلسوف می‌تواند در حال خوردن، خوابیدن (انتظار) یا فکر کردن باشد.
- در صورتی که یک فیلسوف مدت‌زمان طولانی‌ای در انتظار غذا بماند و چیزی به او نرسد، از گرسنگی خواهد مرد.
- فیلسوف‌ها می‌توانند با هم در ارتباط باشند.

الگوریتم نوشته‌شده توسط ما قصد دارد بیشترین تعداد فیلسوف‌ها را، هرچقدر که ممکن است زنده نگه دارد.

برنامه برای اجرا، نیاز به تعدادی پارامتر مشخص دارد که در زیر آورده شده‌اند:

`number_of_philosophers`: تعداد فیلسوفانی که دور میز نشسته‌اند.

`time_to_die`: مدت‌زمانی که فیلسوف می‌تواند انتظار غذا را بکشد. این زمان در مقیاس میلی‌ثانیه دریافت می‌شود و اگر در طول این مدت غذایی به فیلسوف نرسد، از گرسنگی خواهد مرد.

`time_to_eat`: عددی که نشان‌دهنده‌ی مدت‌زمانی است که طول می‌کشد تا فیلسوف غذای خود را تمام کند. این عدد در مقیاس میلی‌ثانیه دریافت می‌شود و در طول این مدت، فیلسوف هر دو چنگال را در دست دارد.

`time_to_sleep`: مدت‌زمانی که فیلسوف در خواب سپری می‌کند. این عدد در مقیاس میلی‌ثانیه دریافت می‌شود.

number_of_times_each_philosopher_must_eat: آرگومانی اختیاری که نشان دهنده تعدادِ مرگ قابل قبول برای تمام فیلسوف‌ها در طول اجرای این برنامه است. اگر این آرگومان را به برنامه ندهیم، شبیه‌سازی بدونِ در نظر گرفتن این مرگ‌ها به کار خود ادامه خواهد داد.

خروجی برنامه، پیامی است که مطابق الگوی زیر به کاربر نشان داده خواهد شد:

```
[timestamp_in_ms] [X] has taken a fork
```

```
[timestamp_in_ms] [X] is eating
```

```
[timestamp_in_ms] [X] is sleeping
```

```
[timestamp_in_ms] [X] is thinking
```

```
[timestamp_in_ms] [X] died
```

برای اجرای برنامه لازم است که ابتدا به پوشه‌ای که برنامه در آن قرار دارد رجوع کرده و سپس دستور زیر را در کامندلاین اجرا کنیم تا برنامه کامپایل شود:

```
make BUILD=pretty
```

حال می‌توانید برنامه را با آرگومان‌های دلخواه خود به این شکل اجرا کنید:

```
./philo_bonus <number_of_philosophers> <time_to_die> <time_to_eat> <time_to_sleep> [number_of_times_each_philo_must_eat]
```

نکته: بازه‌ی قابل قبول برای اعداد داده‌شده به برنامه از 0 تا INT_MAX است.

نمونه‌ای از اجرای این برنامه را در زیر مشاهده می‌کنید:

```
/philo_bonus 4 800 200 200 5
```

پایان.