

نجات سیاره آرگوس

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

در کهکشان دوردستی، سیاره آرگوس تحت حمله روبات‌های هوشمند قرار گرفته است. این روبات‌ها از یک هسته مرکزی کنترل می‌شوند که در قلعه‌ای مستحکم قرار دارد. قهرمانان داستان، تیم افسانه‌ای "سه تفنگدار کهکشانی" به فرماندهی پروفیسور مهدوی، ماموریت دارند تا هسته مرکزی را خاموش کنند و سیاره را نجات دهند.

اما مشکل اینجاست: هسته مرکزی تنها با یک کلید عددی ویژه که از یک الگوریتم پیچیده پیروی می‌کند، خاموش می‌شود. این الگوریتم باید توسط سه سیستم جداگانه (ترد) اجرا شود و هر کدام از این سیستم‌ها بخشی از عملیات را بر عهده دارند.

توضیح ماموریت:

۱. ابتدا هر سه سیستم به ترتیب کار خود را آغاز می‌کنند.
۲. مقدار اولیه عددی برابر با صفر است. هر ترد باید این عدد را دریافت کند، آن را یک واحد افزایش دهد و مقدار جدید را به اشتراک بگذارد.
۳. این عملیات ادامه دارد تا مقدار عدد به 100 برسد.
۴. در نهایت، مشخص می‌شود که هر سیستم چند بار مقدار عدد را افزایش داده است.

ماموریت شما:

به تیم سه تفنگدار کهکشانی کمک کنید تا با نوشتن یک برنامه، این عملیات را به درستی انجام دهند و هسته مرکزی را خاموش کنند.

#ورودی

این برنامه نیازی به ورودی ندارد. مقدار اولیه عدد صفر است و عملیات به صورت خودکار شروع می‌شود.

#خروجی

برنامه باید در نهایت تعداد دفعاتی که هر ترد عدد را افزایش داده است، نمایش دهد.

مثال

در اینجا یک نمونه برای فهم بهتر صورت سوال و قالب خروجی تست‌ها داده می‌شود.

##خروجی نمونه 1

Thread 1 : 31

Thread 2 : 48

Thread 3 : 21

پیروزی نهایی: با اجرای این برنامه، هسته مرکزی خاموش می‌شود، روبات‌ها از کار می‌افتند و سیاره آرگوس بار دیگر به آرامش می‌رسد. نیروهای تاریکی شکست می‌خورند و کهکشان از تیم قهرمان قدردانی می‌کند.

رهای آگوس (نسخه پیشرفته)

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

پس از شکست دادن نیروهای تاریکی و خاموش کردن هسته مرکزی، تیم افسانه‌ای "سه تفنگدار کهکشانی" دوباره با چالشی تازه روبه‌رو می‌شود. هسته مرکزی توسط روبات‌های پیشرفته‌تر بازسازی شده و این بار با الگوریتم‌های جدیدی محافظت می‌شود. برای نجات سیاره آگوس، تیم باید از فناوری‌های پیشرفته‌تر استفاده کند.

این بار، سه تفنگدار کهکشانی باید از سیستم QThread، فناوری پیشرفته‌ای از کهکشان کوانتوم، برای اجرای عملیات استفاده کنند.

**توضیح ماموریت:*

۱. ابتدا سه QThread باید به طور همزمان شروع به کار کنند.

۲. مقدار اولیه عدد برابر با صفر است. هر QThread باید این عدد را دریافت کند، آن را یک واحد افزایش دهد و مقدار جدید را به اشتراک بگذارد.

۳. این عملیات ادامه دارد تا مقدار عدد به 100 برسد.

۴. در نهایت، مشخص می‌شود که هر QThread چند بار مقدار عدد را افزایش داده است.

**ماموریت شما:*

به تیم سه تفنگدار کهکشانی کمک کنید تا با استفاده از QThread این عملیات را به درستی اجرا کنند و سیاره آگوس را نجات دهند.

#ورودی

این برنامه نیازی به ورودی ندارد. مقدار اولیه عدد صفر است و عملیات به صورت خودکار شروع می‌شود.

#خروجی

برنامه باید در نهایت تعداد دفعاتی که هر QThread عدد را افزایش داده است، نمایش دهد.

مثال

در اینجا یک نمونه برای فهم بهتر صورت سوال و قالب خروجی تست‌ها داده می‌شود.

##خروجی نمونه 1

QThread 1 : 31

QThread 2 : 48

QThread 3 : 21

پیروزی نهایی: با اجرای این برنامه، هسته مرکزی برای همیشه خاموش می‌شود، روبات‌های هوشمند از کار می‌افتند و آرامش به سیاره آرگوس بازمی‌گردد. کهکشان بار دیگر با افتخار تیم سه تفنگدار کهکشان را تحسین می‌کند.

رمزگشایی سیگنال‌های گمشده

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

در کهکشان "آندرومدا"، ایستگاه تحقیقاتی "سیگما" تحت حمله نیروهای ناشناخته قرار گرفته است. این نیروها با ارسال سیگنال‌های مخرب، سیستم‌های ارتباطی و دفاعی ایستگاه را مختل کرده‌اند. پروفیسور مهدوی و تیم افسانه‌ای "سه تفنگدار کهکشانی" باید این سیگنال‌ها را ردیابی و رمزگشایی کنند تا سیستم‌های ایستگاه بازگردند.

تکنولوژی اصلی که برای این عملیات نیاز است، سیستم **Signal & Slot** پیشرفته است که در فناوری Qt توسعه یافته است.

**توضیح ماموریت:*

۱. سه دستگاه (یا سامانه) وجود دارند که سیگنال‌های مختلفی را ارسال می‌کنند. هر دستگاه یک پیام خاص (مثلاً یک عدد) ارسال می‌کند.

۲. یک سیستم مرکزی وجود دارد که پیام‌های ارسال‌شده را دریافت می‌کند و آنها را پردازش می‌کند.

۳. با هر پیام دریافت‌شده:

○ سیستم مرکزی باید عدد ارسال‌شده را به مقدار فعلی خود اضافه کند.

○ سپس مقدار جدید پردازش‌شده را نمایش دهد.

۴. عملیات ادامه دارد تا زمانی که مقدار عدد در سیستم مرکزی به 100 برسد.

**ماموریت شما:*

برنامه‌ای طراحی کنید که از مکانیزم **Signal & Slot** استفاده کند:

• هر دستگاه از طریق **Signal** پیام خود را ارسال کند.

• سیستم مرکزی پیام‌های دریافت‌شده را از طریق **Slot** پردازش کرده و مقدار کل را به‌روزرسانی کند.

جزئیات پیاده‌سازی:

۱. کلاس Device:

- این کلاس یک سیگنال به نام `sendSignal` دارد که یک عدد ارسال می‌کند.
- این عدد نشان‌دهنده پیامی است که دستگاه تولید کرده است.
- تابعی عمومی به نام `generateMessage()` برای ارسال سیگنال تعریف کنید.

۲. کلاس CentralSystem:

- این کلاس یک اسلات به نام `processSignal(int number)` دارد که عدد ارسال‌شده را دریافت و به مجموع عددها اضافه می‌کند.
- اگر مجموع عددها به 100 برسد، پیامی مبنی بر پایان عملیات نمایش دهد.

۳. ارتباط Signal و Slot:

- هر دستگاه باید به سیستم مرکزی متصل شود تا سیگنال‌های خود را ارسال کند.

#ورودی

نیازی به ورودی خاص نیست. برنامه به صورت خودکار پیام‌ها را ارسال و پردازش می‌کند.

#خروجی

برنامه باید هر بار که پیامی پردازش می‌شود، مقدار کل را نمایش دهد و در نهایت پایان عملیات را اعلام کند.

مثال

در اینجا یک نمونه برای فهم بهتر صورت سوال و قالب خروجی تست‌ها داده می‌شود.

##خروجی نمونه 1

```
Device 1: Sent message 10   Central system: Total value = 10
Device 2: Sent message 15   Central system: Total value = 25
Device 3: Sent message 20   Central system: Total value = 45
...
Central system: Operation completed with total value = 100.
```

پیروزی نهایی: با این عملیات، پروفیسور مهدوی و تیمش موفق به رمزگشایی سیگنال‌ها می‌شوند و سیستم‌های دفاعی و ارتباطی ایستگاه سیگما دوباره فعال می‌شوند. ایستگاه از حمله نجات پیدا می‌کند و کهکشان آندرومدا بار دیگر به آرامش بازمی‌گردد.

رمزگشایی قفل کهکشانی با فناوری ایستا

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

در اعماق کهکشان "دومینوس"، تیم پروفسور مهدوی با قفلی مواجه می‌شوند که تنها با یک الگوریتم پیچیده و هماهنگی دقیق باز می‌شود. این قفل دارای یک سیستم تحلیل رمز پیشرفته است که از "حافظه ایستا" و "فضای نام اختصاصی" برای مدیریت کدها و الگوریتم‌ها استفاده می‌کند.

ماموریت تیم: پروفسور مهدوی و تیمش باید الگوریتمی طراحی کنند که:

۱. با استفاده از یک **namespace** عملیات ریاضی پیچیده را مدیریت کند.
۲. از یک **static member** برای نگهداری شمارش تعداد قفل‌های باز شده توسط تیم استفاده کند.
۳. الگوریتم اصلی در کلاس `GalaxyUnlocker` پیاده‌سازی شود.

جزئیات ماموریت:

۱. namespace MathOperations:

- شامل دو تابع عمومی:
 - `add(int a, int b)` برای جمع دو عدد
 - `multiply(int a, int b)` برای ضرب دو عدد
- یک تابع خصوصی که فقط در داخل فضای نام قابل استفاده است:
 - `power(int base, int exponent)` برای محاسبه توان عدد

۲. GalaxyUnlocker کلاس:

- شامل یک عضو ایستا (`static`) به نام `unlockedCount` که تعداد قفل‌های باز شده را نگهداری می‌کند.
- شامل تابعی عمومی به نام `unlockCode(int base, int exponent)` که با استفاده از فضای نام `MathOperations` ، توان عدد را محاسبه کرده و بررسی می‌کند آیا این عدد با الگوریتم خاصی

قفل را باز می‌کند یا خیر.

- اگر قفل باز شود، `unlockedCount` افزایش پیدا می‌کند.

۳. شرایط باز کردن قفل:

- عدد تولیدشده باید ضربی از 7 باشد.
- اگر قفل باز نشد، پیام شکست نمایش داده شود.

۴. ورودی:

- عدد پایه (`base`) و توان (`exponent`) به عنوان ورودی دریافت می‌شوند.

وظیفه شما: برنامه‌ای بنویسید که:

- از فضای نام `MathOperations` برای انجام عملیات ریاضی استفاده کند.
- تعداد قفل‌های باز شده را با استفاده از متغیر `static` نمایش دهد.

#ورودی

یک خط شامل دو عدد صحیح `base` و `exponent`.

#خروجی

برنامه باید وضعیت هر تلاش برای باز کردن قفل و تعداد قفل‌های باز شده را نمایش دهد.

مثال

در اینجا یک نمونه برای فهم بهتر صورت سوال و قالب خروجی تست‌ها داده می‌شود.

##ورودی نمونه 1

3 4

2 5

7 2

##خروجی نمونه 1

Base: 3, Exponent: 4 -> Result: 81 -> Lock did not open.
Base: 2, Exponent: 5 -> Result: 32 -> Lock did not open.
Base: 7, Exponent: 2 -> Result: 49 -> Lock opened!
Number of locks opened: 1

***پیروزی نهایی:** استفاده از این الگوریتم، تیم پروفیسور مهدوی موفق به باز کردن دروازه رمزنگاری شده می‌شوند و اطلاعات ارزشمندی برای نجات کهکشان به دست می‌آورند. قهرمانان ما بار دیگر تاریخ‌ساز می‌شوند.

تسخیر قلعه سایه‌ها

• محدودیت زمان: ۱ ثانیه

• محدودیت حافظه: ۲۵۶ مگابایت

در آخرین مأموریت خود، تیم پروفیسور مهدوی باید وارد "قلعه سایه‌ها" شوند؛ مکانی افسانه‌ای که در آن قفل‌هایی وجود دارد که تنها با هماهنگی دانش پیشرفته برنامه‌نویسی باز می‌شوند. این قفل‌ها از ترکیبی از `namespace`، `static members`، `Signal/Slot`، `Threads` و حتی **الگوریتم‌های بازگشتی** برای حفظ امنیت استفاده می‌کنند.

مأموریت: شما باید برنامه‌ای بنویسید که این قفل را باز کند. برای این کار، باید چندین مؤلفه را با هم ترکیب کنید.

جزئیات مأموریت

۱. namespace LockUtils:

- شامل دو تابع عمومی:
 - `isPrime(int number)` که بررسی می‌کند آیا عدد اول است یا خیر
 - `gcd(int a, int b)` که بزرگ‌ترین مقسوم‌علیه مشترک دو عدد را محاسبه می‌کند
- یک تابع خصوصی که فقط در فضای نام استفاده می‌شود:
 - `factorial(int n)` که فاکتوریل عدد را محاسبه می‌کند

۲. Lock کلاس:

- شامل یک متغیر ایستا (`static`) به نام `unlockCount` برای شمارش تعداد قفل‌های باز شده
- شامل سیگنالی به نام `attemptUnlock(int code)` که کدی را برای باز کردن قفل ارسال می‌کند
- شامل تابع عمومی `validateCode(int code)` که با استفاده از الگوریتم‌های موجود در `LockUtils` بررسی می‌کند آیا قفل باز می‌شود یا خیر

۳. CentralControl کلاس:

- اسلاتی به نام `processCode(int code)` دارد که سیگنال `attemptUnlock` را دریافت می‌کند.
- این اسلات باید کد را بررسی کرده و در صورت موفقیت پیام "قفل باز شد" و در صورت شکست پیام "قفل باز نشد" نمایش دهد.

۴. برای تولید کدها Thread:

- موازی وجود دارند که هر کدام کدهای عددی تصادفی تولید می‌کنند و آنها را به Thread سه.
- کلاس `Lock` ارسال می‌کنند.
- آنها باید از `QThread` استفاده کنند و هر کدام 10 کد تولید کنند Thread این.

۵. شرایط باز شدن قفل:

- عدد باید اول باشد.
- عدد باید ضربی از 5 باشد.
- عدد باید با کد خاصی (مثلاً بزرگ‌ترین مقسوم‌علیه مشترک 120 و 45) برابر باشد.

۶. خروجی:

- هر بار تلاش برای باز کردن قفل باید ثبت شود.
- در پایان، تعداد کل قفل‌های باز شده و زمان کل عملیات نمایش داده شود.

#ورودی

- نیازی به ورودی خاص نیست. کدها به صورت تصادفی توسط Thread ها تولید می‌شوند.

#خروجی

برنامه باید وضعیت هر تلاش برای باز کردن قفل و تعداد قفل‌های باز شده را نمایش دهد.

مثال

در اینجا یک نمونه برای فهم بهتر صورت سوال و قالب خروجی تست‌ها داده می‌شود.

##خروجی نمونه 1

Thread 1: Code 17 -> Lock did not open.Thread 2: Code 25 -> Lock did not open.

پیروزی نهایی: با باز شدن قفل‌های قلعه سایه‌ها، تیم پروفیسور مهدوی به گنجینه‌ای از دانش دست پیدا می‌کند که کلید تغییر آینده بشریت خواهد بود. این مأموریت، نقطه اوج ماجراجویی‌های آنهاست.