

نام دانشجو:

شماره دانشجویی:

سوال ۱. مهرداد به تازگی به تیم توسعه شرکت Yelp پیوسته است. این شرکت امکان جستجو، ثبت نظر و ارزیابی کسب و کارهای محلی را فراهم می‌کند. برای مدل‌سازی موقعیت مکانی، تیم توسعه از روش Geohash استفاده می‌کند. به عنوان وظیفه اولیه خود، مهرداد باید موقعیت مکانی tnke2 را تعیین کند و جدول زیر را تکمیل کند. با این حال، به او گفته شده است که متاسفانه تعداد رستوران‌های موجود در ناحیه tnke2 در هنگام جستجو در سایت کافی نیست (کمتر از ۲۰ تا) و باید به دنبال رستوران‌های موجود در نواحی اطراف نیز بگردد تا تعداد کافی را برای فراهم کردن تجربه کاربری بهتر برای مشتریان بدست آورد. با فرض استفاده از Geohash، چه راهکاری را برای حل این مشکل به مهرداد پیشنهاد می‌دهید؟

Base 32	t	n	k	e	2
Decimal	29			14	2
Binary	11101			01110	00010
Longitude (Binary)					
Latitude (Binary)					
Longitude (Degree)					
Latitude (Degree)					

سوال ۲. مرتضی به عنوان یک مهندس نرم‌افزار در آمازون فعالیت می‌کند. یکی از قابلیت‌های مرتبط با جستجو، کارایی مناسبی ندارد و از مرتضی خواسته شده است که این مشکل را حل کند. این قابلیت به سازمان‌ها اجازه می‌دهد تا بین سفارشات که به مشتریان تحویل داده شده است، جستجو کنند. این جستجو بر اساس نوع کالا، زمان خرید و کشوری که کاربر در آن زندگی می‌کند، انجام می‌شود. همچنین، امکان فیلتر کردن نتایج جستجو بر اساس تعدادی کلیدواژه نیز میسر است. به عنوان مثال، شرکت اپل می‌خواهد بداند که در سال ۲۰۲۳، چه افرادی از ایالت کالیفرنیا در آمریکا، گوشی آیفون ۱۵ را از آمازون خریداری کرده و پرداخت در محل انجام داده‌اند. پس از بررسی، مرتضی متوجه مشکل اصلی شد. در حال حاضر، برای پیاده‌سازی سیستم، از معماری مایکروسرویس استفاده می‌شود، به طوری که هر سرویس پایگاه داده مستقل خود را دارد و برای این جست‌وجو نسبتاً پیچیده، به هر سرویس یک کوئری ارسال می‌شود و منطق برنامه و تجمیع آن‌ها در سطح کد انجام می‌شود. این موضوع باعث کاهش کارایی شده است. با این حال، مرتضی اخیراً الگوهای ابری را یاد گرفته و باور دارد که راه حل بهتری برای این مسئله وجود دارد. به نظر شما، راه حل مرتضی چیست؟ با رسم یک شکل، توضیح دهید.

سوال ۳. بهمن به تازگی به عنوان یک آزمونگر نرم افزار در شرکت فیسبوک استخدام شده است. او روش جدیدی برای ارزیابی اثربخشی تست ها با نام آزمون موتاسیون را یاد گرفته است. این روش بر اساس ایجاد مجموعه ای از خطاهای مصنوعی در کد برنامه است. در فرآیند آزمون موتاسیون، تغییرات نحوی کوچکی در سورس کد برنامه ایجاد می شود که این تغییرات به عنوان موتانت شناخته می شوند. مثلاً در کد زیر می توان موتانت های زیر را تولید کرد:

- موتانت M1: در برنامه اصلی، عملگر % در خط ۴ با عملگر > جایگزین می شود.
- موتانت M2: در برنامه اصلی، عملگر % در خط ۴ با عملگر < جایگزین می شود.
- موتانت M3: در برنامه اصلی، عملگر % در خط ۴ با عملگر == جایگزین می شود.
- موتانت M4: در برنامه اصلی، عملگر % در خط ۴ با عملگر >= جایگزین می شود.
- موتانت M5: در برنامه اصلی، عملگر % در خط ۴ با عملگر <= جایگزین می شود.

برای هر موتانت، تمام تست های برنامه اجرا می شوند. اگر تست های برنامه به اندازه کافی خوب باشند، انتظار می رود حداقل یک تست شکست بخورد. در این حالت، می گوییم موتانت کشته شده است. اگر تمام تست ها موفق باشند، این به این معنی است که تست ها به اندازه کافی خوب نبوده اند و موتانت زنده مانده است. در نهایت، می توان یک عدد بین صفر و یک را به عنوان پوشش آزمون موتاسیون به دست آورد که برابر است با تعداد موتانت های کشته شده تقسیم بر تعداد کل موتانت ها. وضعیت هر موتانت برای برنامه زیر را با توجه به تست نوشته شده مشخص کنید. علاوه بر این، میزان پوشش آزمون موتاسیون را برای تست فعلی محاسبه کنید. چه تست های جدیدی باید اضافه شوند تا میزان پوشش آزمون موتاسیون کد زیر را به حداکثر برسانیم؟

```
def sum_of_odd_numbers(list_):
```

```
    total = 0
```

```
    for element in list_:
```

```
        if element % 2:
```

```
            total += element
```

```
    return total
```

```
# Test Cases
```

```
T1: sum_of_odd_numbers([1,2])
```
