



دانشکده مهندسی کامپیوتر

آزمون نرم افزار (۴۰۸۲۸)

استاد: دکتر حسن میریان

تمرین اول

امیرحسین کارگران خوزانی (۹۹۲۰۱۱۱۹)

سید سجاد میرزا بابایی (۹۹۲۱۰۱۴۲)

رامتین باقری (۹۹۳۰۱۹۳۸)

بخش نظری

۱. در این بخش باید یک برنامه تبدیل اعداد رومی به اعداد دهدهی را مورد بررسی قرار دهید. این برنامه به زبان برنامه نویسی جاوا نوشته شده است.

(a) خطا یا خطاهای برنامه در کدام قسمت هستند؟ اصلاح شده آنها را بنویسید.

خروجی مورد انتظار این تابع برای ورودی II مقدار 2 خواهد بود؛ اما پس از اجرای تابع با ورودی مذکور، مقدار 0 نتیجه می شود. این مشکل از آنجا سرچشمه می گیرد که افزودن عدد جدید به اعداد قبلی تنها در صورتی انجام می شود که شرط `currentNumber > next` برقرار باشد، با این حال نیاز است تا برابری این دو متغیر نیز برای افزودن عدد لحاظ شود. با تغییر در خط ۲۳ و اضافه کردن حالت برابری دو متغیر، مشکل حل خواهد شد. نسخه اصلاح شده برنامه در قسمت موجود است.

(b) در صورت امکان مورد آزمونی ارائه دهید که خطا را اجرا نکند.

از آنجایی محل خطا در خط شماره ۲۳ قرار داد، باید مورد آزمون را به نحوی طراحی کنیم که این خط از کد اجرا نشود. از این رو مورد آزمون هدف، دارای ورودی "" (رشته متنی با طول صفر) و خروجی قابل انتظار 0 است.

(c) در صورت امکان آزمونی بنویسید که خطا را اجرا کند اما نتیجه حالت میانی و پایانی مشخص کننده حالت اشتباه نباشد.

همانطور که بالاتر گفته شد، با اجرا شدن خط شماره ۲۳، خطا نیز اجرا خواهد شد. مورد آزمون هدف، ورودی IV و خروجی قابل انتظار 4 را خواهد داشت. در این صورت خطا اجرا می شود ولی تاثیری بر فرآیند محاسبه نخواهد گذاشت.

(d) در صورت امکان آزمونی بنویسید که برنامه در حالت میانی اشتباه است اما در پایان نتیجه شکست نمی شود. با مثالی نتیجه مورد انتظار و نتیجه اجرا را نشان دهید.

برنامه زمانی دارای حالت میانی اشتباه می شود که

۱. خط شماره ۲۳ اجرا شود، و

۲. ورودی دارای حداقل دو کاراکتر یکسان متوالی باشد.

با بررسی برنامه متوجه شدیم که اگر دو کاراکتر یکسان متوالی قبل از یک کاراکتر با ارزش تر قرار گیرند، علی رغم اشتباه بودن حالت میانی در حین محاسبه، نتیجه نهایی درست خواهد بود. برای مثال، در هنگام اجرا با مورد آزمون با ورودی IIX و خروجی قابل انتظار 12، مقدار `currentNumber` به 2- نیز خواهد رسید که مشخصاً یک حالت میانی اشتباه است. با این حال، بعد از اتمام برنامه برای ورودی مذکور، خروجی واقعی برابر با خروجی قابل انتظار خواهد بود.

(e) با استفاده از تحلیل آر.آی.پی شرایطی را تعیین کنید که مورد آزمون تشخیص دهنده خطای این برنامه باید داشته باشد.

نتیجه تحلیل **قسمت** با استفاده از مدل RIP، به شکل زیر است:

$ S > 0$	دسترسی پذیری
$\exists i, 0 \leq i < i+1 < S \wedge S[i] = S[i+1]$	آلودگی
$\exists i, 0 \leq i < i+1 < i+2 < S \wedge S[i] = S[i+1] \wedge C(S[i+1]) < C(S[i+2])$	انتشار

بنابراین، مشخصات آزمون تشخیص دهنده خطای برنامه عبارت است از:

$$|S| > 0 \wedge$$

$$(\exists i, j, 0 \leq i < j < |S| \wedge j - i = 1 \wedge S[i] = S[j]) \wedge$$

$$(\exists i, 0 \leq i < i+1 < i+2 < |S| \wedge S[i] = S[i+1] \wedge C(S[i+1]) < C(S[i+2]))$$

۲. برنامه موجود در **جدول** کوچکترین و بزرگترین عنصر آرایه‌ای از اعداد صحیح را پیدا می‌کند. پیاده‌سازی این برنامه اشتباه است. با استفاده از روش افراز فضای ورودی، موارد آزمونی را طراحی کنید که خطای برنامه را مشخص کند. رویکرد شما باید حداقل دو خصوصیت مبتنی بر عملکرد و یک خصوصیت مبتنی بر واسط داشته باشد.