# گزارش ارزیابی مجموعه PrintTokens2

**تعداد ورژن­های خطادار = 10**

ورژن 1 – خطوط 189 تا 194 از سورس اصلی حذف شده­اند.

سوال؟ وقتی چند خط حذف می­شود چجوری پردیکت کنیم؟

بعد از مستندگذاری مسیرهای P111 و P59 عامل خطا در ورژن 1 هستند.

در نتایج خطا P59 شناسایی شده است. موردی که وجود دارد این است که هیچ کدام از تست کیس ها خط P111 را پوشش نداده اند.

ورژن 2 – if خط 176 سورس اصلی حذف شده است.

بعد از مستندگذاری مسیرهای P111 عامل خطا در ورژن 2 است.

ورژن 3 – خط 177 سورس اصلی حذف شده است  
بعد از مستندگذاری اگر P29 اجرا شود خط کامنت شده پوشش داده می­شود عامل خطا در ورژن 3 است.

ورژن 4 – خط 164 سورس اصلی یک مقدار یک انتساب تغییر کرده است.

بعد از مستندگذاری با توجه به وجود خطا در یک مقدار ثابت در هدر فایل مسیر P69 با توجه به شرط می­تواند عامل خطا باشد یا نباشد.

ورژن 5 – خط 386 سورس اصلی یک مقدار bool به TRUE تغییر کرده است.

بعد از مستندگذاری با توجه به کامنت شدن یک خط از کد اگر مسیر P32 اجرا شود خط کامنت شده پوشش داده می­شود که عامل خطا در ورژن 5 است.

ورژن 6 – خط 358 سورس اصلی یک مقدار به یک متغیر اضافه شده است.  
بعد از مستندگذاری با توجه به وجود خطا در چند مقدار ثابت در هدر فایل، باید هر کجا مقادیر در فایل cpp استفاده شده است زنجیره رخداد خطا را بررسی کرد.

ورژن 7 – خط 218 سورس اصلی به شرط کد یک شرط اضافه شده است.

بعد از مستندگذاری مسیرهای P32 عامل خطا در ورژن 2 است.

ورژن 8 – خط 225 سورس اصلی به شرط کد یک شرط اضافه شده است.

بعد از مستندگذاری مسیرهای P32 عامل خطا در ورژن 2 است.

ورژن 9 – خط 218 سورس اصلی به شرط کد یک شرط اضافه شده است.

بعد از مستندگذاری مسیرهای P32 عامل خطا در ورژن 2 است.

ورژن 10 – خط 380 سورس اصلی یک مقدار از درون شرط while تغییر کرده است.

بعد از مستندگذاری مسیرهای P32 عامل خطا در ورژن 2 است.

سوال وقتی یک لاین از کد پاک شود چطور خطا مشخص می­شود؟

بعد از مستندگذاری کدها و اجرای کدهای مستندگذاری شده فایل‌های خروجی حاصل از مستندگذاری نشان می‌داد که برخی از تست­ ها دقیقاً مسیرهای یکسانی را طی می‌کنند که میتوان تست های مشابه را حذف و یک نمونه از آن‌ها باقی گذاشت. این جر مشکلات زیمنس می تونه مطرح بشه

ایده قناتی: بعد از شناسایی اجراهای تصادفاً موفق بیایم به جای حذف اونا از اونا استفاده کنیم و با توجه به مسیرهای که تصادفاً موفق­ها رفتند سعی کنیم مسیری ک خطا دار است رو پیدا کنیم. چطوری؟ دکتر پیشنهاد داد تست دیتا بدیم.  
ایده خود من اینه که با اجراهای صحیحی که با اونا تو یه کلاستر قرار گرفتن شروع کنیم تست دیتاهای مشابه اون کلاستر رو تولید کنیم تا بتونیم مرز تفاوت ناموفق ها و تصادفاً موفق ها رو پیدا کنیم که این مرز رو همون بازی با تست دیتاها مشخص می کنه. برای اینکه یادم نره به فرض شرط x>20 به اشتباه x>30 نوشته شده است. خب اینجا مثلا x=36 بدهیم واقعا جواب ما درست است و اگر ورودی در بازه­ی 20-30 باشد اجراهای با این ورودیا نتیجه غلط می­دهند. با مقایسه اینا یه جوری میشه تست دیتایی که باعث ایجاد مرز بین ناموفق ها و تصادفاً موفق ها میشه رو پیدا کرد. درود اقا میثم برو ببینم چ میکنی