# گزارش ارزیابی مجموعه PrintTokens2

**تعداد ورژن­ های خطادار = 10**

# ورژن 1

### شرح خطا:

با مستندگذاری سورس اصلی و سورس خطادار و مقایسه دو کد متوجه شدیم که یک شرط با بدنه شرط از کد حذف شده است.

محل رخداد خطا:

خطا با اجرای P44 پوشش داده شده است.

نتیجه­ ی اجرا:

# ورژن 2

### شرح خطا:

با مستندگذاری سورس اصلی و سورس خطادار و مقایسه دو کد متوجه شدیم که یک شرط از کد حذف شده است اما بدنه شرط باقی‌مانده است.

محل رخداد خطا:

خطا با اجرای P44, P45 پوشش داده شده است.

نتیجه ­ی اجرا:

# ورژن 3

### شرح خطا:

با مستندگذاری سورس اصلی و سورس خطادار و مقایسه دو کد متوجه شدیم که یک خط از کد حذف شده است.

محل رخداد خطا:

خطا با اجرای P37 پوشش داده شده است.

نتیجه ­ی اجرا:

# ورژن 4

### شرح خطا:

با مستندگذاری سورس اصلی و سورس خطادار و مقایسه دو کد متوجه شدیم که مقدار یک انتساب در بدنه یک شرط تغییر کرده است.

محل رخداد خطا:

خطا با اجرای P31,P32 پوشش داده شده است.

نتیجه ­ی اجرا:

# ورژن 5

### شرح خطا:

با مستندگذاری سورس اصلی و سورس خطادار و مقایسه دو کد متوجه شدیم که یک مقدار bool از FALSE به TRUE تغییر کرده است.

محل رخداد خطا:

خطا با اجرای P121 پوشش داده شده است.

نتیجه ­ی اجرا:

# ورژن 6

### شرح خطا:

با مستندگذاری سورس اصلی و سورس خطادار و مقایسه دو کد متوجه شدیم که یک خطا با اضافه شدن یک مقدار به یک متغیر درون یک شرط در کد کشت شده است.

محل رخداد خطا:

خطا با اجرای P110,P111 پوشش داده شده است.

نتیجه ­ی اجرا:

# ورژن 7

### شرح خطا:

با مستندگذاری سورس اصلی و سورس خطادار و مقایسه دو کد متوجه شدیم که یک خطا با اضافه شدن یک شرط به شرط دیگر در کد کشت شده است.

محل رخداد خطا:

خطا با اجرای P55, P56 پوشش داده شده است.

نتیجه ­ی اجرا:

# ورژن 8

### شرح خطا:

با مستندگذاری سورس اصلی و سورس خطادار و مقایسه دو کد متوجه شدیم که یک خطا با اضافه شدن یک شرط به شرط دیگر در کد کشت شده است.

محل رخداد خطا:

خطا با اجرای P61 پوشش داده شده است.

نتیجه ­ی اجرا:

# ورژن 9

### شرح خطا:

با مستندگذاری سورس اصلی و سورس خطادار و مقایسه دو کد متوجه شدیم که یک خطا با اضافه شدن یک شرط به شرط دیگر در کد کشت شده است.

محل رخداد خطا:

خطا با اجرای P55, P56 پوشش داده شده است.

نتیجه ­ی اجرا:

# ورژن 10

### شرح خطا:

با مستندگذاری سورس اصلی و سورس خطادار و مقایسه دو کد متوجه شدیم که یک خطا با تغییر یک مقداردرون یک شرط while در کد کشت شده است.

محل رخداد خطا:

خطای ایجاد شده در ورژن 10 موقع اجرا با تست دیتاها باعث ایجاد segmentation\_fault و core\_dump می­شود.

خطا با اجرای P116, P117 پوشش داده شده است.

نتیجه ­ی اجرا:

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

سوال؟ وقتی چند خط حذف می­شود چجوری پردیکت کنیم؟  
بعد از مستندگذاری مسیر P111 مشخص کننده محدوده خطوط حذف شده از کد است و اگر این پریدیکت دیده شود یعنی خطا اجرا شده است.

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 با توجه به حذف شدن یک شرط P37 نماینده خطا است.

ورژن 3 – خط 177 سورس اصلی حذف شده است  
بعد از مستندگذاری اگر P32 اجرا شود خط حذف 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 اگر P26 اجرا شود خطا پوشش داده می­شود.

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

به به P85 عامل خطاست.

ورژن 6 – خط 358 سورس اصلی یک مقدار به یک متغیر اضافه شده است.  
عامل خطا که درون شرط است بین P77 و P78 قرار دارد.

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

خطا بین P41 و P45 رخ خواهد داد.

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

خطا بین P41 و P48 رخ خواهد داد.

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 مسیر P82 عامل انتشار خطاست.

خطای ایجاد شده در ورژن 10 موقع اجرا با تست دیتاها باعث ایجاد segmentation\_fault و core\_dump می­شود.

توی ورژن 10 توی اینسترومنت دیدم خیلی اینسترومنتمون روی شرطها درست نیست وگرنه هر چقد اینسترومنت دقیق تر باشه میزان بررسی بعد از پیش بینی مسیر خطادار توسط کد کمتر خواهد شد.

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

توی مجموعه printtoken2 خطاهایی در سطح شرط و جملات درون if داشتیم که اگر با instrument کلی که داریم ینی در سطح شرط نیست. جواب نگرفتیم می توان با instrument سطح شرط دقت مکان یابی را بیشتر کرد.  
توی پیشنهادات بگم که دقت روش با instrument سطح شرط خیلی بالاتر خواهد رفت.

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