

Здравейте колеги ☺

Това е кратък списък от правила, които трябва да спазвате, когато предавате задача в система за автоматично тестване. Някои от тях са съвсем общи, а други се отнасят за конкретни системи, които ползваме на курса.

1. Входните данни се четат в абсолютно същия формат, както са описани в задачата и в примерния вход. Гарантирано е, че данните ще идват при вас в правилния формат. В някои задачи whitespaces-овете (интервал, табулация, \n) **ИМАТ** значение. Никой не е длъжен да пише checker-и, които да се абстрахират от липсващи или допълнителни whitespaces-ове.
2. В общия случай не се изисква от вас да правите тестове за коректност на входа, освен ако това изрично се споменава в задачата.
3. Абсолютно недопустимо е да принтите на стандартния изход каквито и да било данни, които не са във формата на изхода. Това включва всякакви подканящи съобщения от типа „*Vuvedi N=*”. Никой никога няма да се занимава да въвежда ръчно вход на вашите програми. За тестване се ползва пренасочване на входа и изхода на програмата ви.
4. Всеки ред се смята за завършен само когато на края му има '\n'. Много хора го наричат символ за нов ред, а всъщност далеч по-правилно би било да бъде наричан символ за край на ред. Целият изход на вашата програма трябва да се състои от завършени редове. Т.е. на края на изхода задължително трябва да присъства така обсъждания символ.
5. Някои колеги прибегват до функции като *system("pause")* за да тестват програмите си, без те да се затварят веднага след като приключат работата. Доста хора забравят да я махнат преди да си пратят решението, което в добрия случай води до RE, а доста по-често до WA.
Решението: Тествайте под конзола или в достатъчно умна среда, която не ви затваря конзолата веднага след приключването на програмата ви. Примерно **Code::Blocks** или **Visual Studio** + един breakpoint на *return 0;* в main-а.
6. Внимавайте за типовете, които ползвате в решението си. Често задачите са нагласени да прехвърлят стандартния целочислен тип `int`, затова ползвайте `long long`, когато прецените за нужно. Понякога се срещат и екстремни случаи, в които стойностите прехвърлят дори 2^{64} . Тогава се налага да пишете „дълги” числа.
7. Внимавайте със системния стек. Твърде дълбоката рекурсия или заделяне на големи масиви могат лесно да го препълнят и да причинят RE.
8. **cin/cout** са **БАВНИ**. Особено силно се усеща при голям вход /големи тестове/ и, разбира се, води до TL. Ползвайте го само по желание за задачи с малки входове, а в другите случаи – **printf/scanf**.

Основни типове резултати, които връща системата:

- OK – ясно е ☺
- WA – wrong answer – грешен отговор
- SG – segmentation fault – „пипате” в чужда памет, погледнете докъде обхождате масивите си, в общия случай помага.
- RE – runtime error – получава се при неправилно поведение на програмата, например делене на 0
- PE – presentation error - получава се, когато решението на състезателя се различава от исканото само с whitespaces-ове. /рядко се поддържа/