



## Десето вежбање

- У развојном окружењу CS50 IDE направити нови директоријум `dan08`.
- За потребе провере усклађености са МИСРА стандардом користиће се компајлер Тексас Инструментс Ц6000 (у даљем тексту ТИ).
  - Преузети алат, распаковати и прекопирати у развојно окружење
  - Позиционирати се на путању где је копирано, у директоријум `ti_tools` и покренути скрипту: `source set_env.sh` (напомена: овај корак је потребно урадити сваки пут када се отвори нови терминал)
  - Скрипта из претходног корака даје права извршавања копираним алатима и поставља системске путање како би се алат могао користити директно са било које путање.
  - Детаљно упутство за коришћење ТИ компајлера налази се овде. Опис команди за покретање компајлера се налази у поглављима 2.2 и 2.3.1, док је опис коришћења МИСРА провере дат у 6.3.
- Позивање ТИ компајлера представља додатни корак у развоју кода. Да би се програм покренуо и даље је потребно превести га са ГЦЦ. Излаз из ТИ компајлера није битан (објектна датотека), већ искључиво испис који представља резултат провере усклађености са МИСРА стандардом.

### Вежба 1

Користећи алат за проверу усклађености са МИСРА стандардом извршити проверу за датотеку `program.c` из Задатка 1 (прво вежбање на курсу).

Алат можете позвати са: `cl6x --include_path=$TI_INCLUDE --compile_only --check_misra=all program.c`

Ако је све постављено како треба, конзолни испис ће бити затрпан извештајем о проблемима са MISRA правилима. Усредсредите се на информације које се тичу датотеке `program.c`, занемарите оне које се односе на системска заглавља. Размислите о сваком нарушеном правилу и о начину како би се оно могло задовољити.



Пробајте да преправите код, тако да буде задовољено што је могуће више MISRA правила. За разумевање правила (и одлуке како их најбоље задовољити) помоћи ће вам следећи документи:

"MISRA-C:2004 - Guidelines for the use of the C language in critical systems" - MISRA 2004 стандард

"MISRA C 2004 Permits" - Списак правила које је дозвољено прекршити и разлог и околности под којима је то дозвољено

"MISRA C 2012 Addendum 1 - Rule Mapping" - Табела пресликавања MISRA 2004 правила на MISRA 2012 правила, са образложењима зашто су нека правила промењена (појачана, релаксирана, избачена или су уведена нова)

У случају да вам горе наведена документа нису доступна, онда бар погледајте списак MISRA 2004 правила, са врло кратким описима у овом документу "IAR Embedded Workbench MISRA C:2004 Reference Guide" овде

У MISRA 2004 стандарду погледајте опис сваког нарушеног правила, а онда пробајте да пронађете то правило и у друга два документа.

Посебну пажњу обратите на следећа правила:

2.2 - Ово правило је у новијој верзији стандарда промењено из обавезног у препоручљиво. Анализирајте разлоге за увођење тог правила и за касније избацивање из скупа обавезних правила. Сами одлучите да ли желите да га се придржавате.

5.7 - Избачено у MISRA 2012 стандарду. Размислите о овом правилу и одлучите да ли ћете га се придржавати.

8.1 - Релаксирано у MISRA 2012 стандарду. Анализирајте шта је промењено у овом правилу између два стандарда. (Фраза „internal linkage” означава да је функција приватна)

17.4 - Релаксирано у MISRA 2012 стандарду и промењено из обавезног у препоручљиво. Анализирајте разлике и размислите о њима. Присетите се различитих начина на које можемо декларисати да функција прима низ као параметар - у томе се крије поштовање овог правила.

20.9 - Да ли је ово правило могуће испоштовати у овом задатку?



# Напредни Це

Попишите сва правила која су иницијално била нарушена у датотеку `spisak_prekršenih_pravila.txt` и поред сваког наведите како сте га решили.

За одређена правила закључак може бити да не желимо или не можемо да их задовољимо.

За поједине делове кода можете искључити МИСРА проверу користећи `#pragma CHECK_MISRA("none")`. Да бисте поново укључили МИСРА проверу потребно је додати `#pragma CHECK_MISRA("all")`. Ово можете искористити да искључите проверу у заглављима (све `#include` директиве).

Проверу за поједина правила можете искључити додавањем `#pragma CHECK_MISRA("-X")` где X представља редни број правила. Поновно укључивање истог правила се врши са `#pragma CHECK_MISRA("X")`.

Више детаља о укључивању/искључивању провере за поједина правила или делове кода можете погледати у упутству (поглавља 6.3, 6.9.1 и 6.9.25).

Задатак је готов када анализатор више не буде пријављивао ни једно нарушено правило у датотеци `program.c`.

Након сваке измене кода обавезно превести код и са компајлером ГЦЦ, покренути и проверити да ли се програм и даље извршава исправно

## Вежба 2

Играјте се мало и анализирајте разне кодове са ранијих вежби. Посебно обратите пажњу на код из претходне вежбе (Кружни бафер).