**1. Minimum koji studenti MORAJU imati je implementirani CRUID.**

**Create-Read-Update-Insert-Delete i tu se misli na kontekst sadržaja koji se zapisuje u datoteku koja može biti tekstualna ili binarna. Dalje samo nadograđujete svoj program!**

**Linija 20 – 86 , utility.c**

**2. Odabir konkretnih primitivnih tipova podataka za rad s cjelobrojnim i realnim konstantama.**

**dataType.h**

**3. Odabir konkretnih složenih tipova podataka za rad sa specifičnim objektima.**

**dataType.h**

**4. Primjena typedef sa strukturama i unijama, po potrebi s enum tipovima podataka tamo gdje treba.**

**dataType.h**

**5. Imenovan<je identifikatora (varijabli, konstanti, polja, funkcija, pokazivača…) – upotreba camelCase, PascalCase i snake\_case konzistento kroz cijeli projekt.**

**6. Primjena ključne riječi static za globalne i lokalne varijable.**

**Linija 114, functions.c**

**7. Organizacija izvornog kôda.**

**8. Primjena extern ključne riječi za globalne varijable.**

**Linija 73, dataType.h**

**Linija 14, utility.c**

**9. Ako su funkcije jednostavne koristiti makro funkcije ili inline funkcije.**

**validDate, validID, validName - utility.c**

**10. Izbornik/podizbornici.**

**menu.c**

**11. Kod izbornika koristiti enum tipove ili makro simbole.**

**12. Generalno upotreba pokazivača tamo gdje su potrebni.**

**13. Generalno upotreba struktura i funkcija.**

**14. Zaštita parametara kod svih funkcija.**

**15. Koristiti statički zauzeta polja gdje su potrebna, nikako ne koristiti VLA polja.**

**Linije 5, 6, dataType.h**

**Linija 63, main.c**

**16. Koristiti dinamičko zauzimanje memorije za bilo koji tip podatka, osobito za složene tipove**

**podataka.**

**Linija 56, utility.c**

**17. Koristiti funkcije malloc(), calloc(), realloc(), free() – neku od njih, ako ne i sve.**

**Linija 29, realloc(), functions.c**

**Linija 51, malloc(), utility.c**

**18. Sigurno brisanje memorije koja je dinamički zauzeta, anuliranje memorijskog prostora, provjera**

**pokazivača kako se ne bi dogodila pogreška double free() i anuliranje svih pokazivača koji su bili**

**usmjereni na memorijski prostor koji se dinamički zauzeo.**

**Linija 51, malloc(), utility.c**

**19. Datoteke, koristiti tekstualnu ili binarnu, provjera pokazivača i zatvaranje datoteke.**

**Linije 43, 44, 81, utility.c**

**20. Koristiti funkcije fseek(), ftell(), rewind(), ovisno o potrebi – neku od njih ako ne sve.**

**Linija 77, rewind(), utility.c**

**21. Koristiti funkcije remove(), rename(), po potrebi implementirati funkciju za kopiranje datoteka.**

**Linija 387, 398, utility.c**

**22. Upravljati s pogreškama, errno, perror(), strerror(), feof(), ferror() – neku od njih ako ne sve.**

**Linija 30, functions.c**

**23. Sortiranje – preporuka koristiti ugrađenu qsort() funkciju.**

**Linija 193, utility.c**

**24. Pretraživanje – preporuka koristiti ugrađenu bsearch() funkciju.**

**Linija 113, functions.c**

**25. Rekurzije je najlakše koristiti primjenom rekurzivnih algoritama sortiranja kao što su quick sort,**

**merge sort, insert sort ili pretraživanja kao što je binary search.**

**Linija 193 - 252, utility.c**

**26. Pokazivače na funkcije je najlakše koristiti upotrebom funkcije qsort() ili bsearch() iz standardne biblioteke.**

**Linija 74, functions.c**