TUNNITÖÖ IAX 0583 28.11.2024 4p+(2p funktsioonikoodid)

*Tänase ülesande eesmärk on luua programmi* ***main() koos funktsioonide väljakutsumistega - ja prototüübid****.* Antud osa ülesande koodist kirjutatakse ilma tehisintellekti või vestlusroti abita, kuna peamine eesmärk on parandada arusaamist programmeerimise alustest. Koodi kirjutamisel tuleb järgida programmide sisendi, töötlemise ja väljundi eraldamise põhimõtet.

Nagu öeldud, on peamine eesmärk luua valitud programmi põhifunktsioon ja funktsioonide prototüübid. See tähendab, et põhifunktsiooni sees kirjutate välja kõik funktsioonikutsed ja muutujate deklaratsioonid. **MAIN()** peaks sisaldama AINULT funktsioonikutseid ja muutujate deklaratsioone. Samamoodi EI TOHI funktsioonide väljakutseid ja muutujate deklaratsioone kirjutada väljaspool main() funktsiooni.

Nende funktsioonide ja muutujate nimede valik jääb programmi koostaja otsustada, kuid see peaks olema seotud funktsiooni või muutuja kasutamisega (nt funktsioon “userInput” kasutajalt sisendi saamiseks või muutuja “i” iteratsiooniks). Funktsioonide nimetamisel peaksid järgima **kas Camelcase'i (funktsiooninimi) või Snake'i (funktsiooni\_nimi**) nimetamise tava. Valige üks kasutamiseks ja **ÄRGE segage neid kahte**. Eesmärk on parandada loetavust ja järjekindel funktsioonide nimetamise tava muudab koodi mõistmise palju lihtsamaks.

Funktsioonide poole pöördumiseks peate looma sobivad parameetrid ja nende kirjeldused prototüüpidesse. **Prototüübid on välja kirjutatud põhifunktsiooni kohale**, kuid allpool teeke. Funktsioonikutsete ja prototüüpide loomisel on hea meeles pidada, millist teavet funktsioonilt soovite saada ja millist teavet funktsiooni andmiseks vajate. See aitab valida andmetüüpe iga funktsiooni prototüübi ja selle parameetrite jaoks. *NÄITEKS:*

*int readSingleInput(char[]) tähendab, et tahan saada funktsioonist return abil int väärtuse ja edastan funktsioonile kasutajale väljatrükkimiseks char massiivi, mis sisaldab võimalikku teadet kasutajale.*

Funktsioonide prototüüpide loomisel peavad olema vähemalt sisendi, töötlemise ja väljundi prototüübid. ***NB! See ei tähenda, et prototüüpe tuleb nimetada sisendiks, töötlemiseks ja väljundiks, vaid seda, et programmi kolm põhiosa on esindatud funktsioonikutsetes ja prototüüpides.*** See tähendab ka, et kolm põhifunktsiooni prototüüpi võivad koosneda mitmest funktsiooni prototüübist (nt readInputInt ja readInputFloat on eraldi funktsiooni prototüübid, kuid esindavad sisendit tervikuna).

Eelmiste nädalate materjalide ja teie varem loodud programmide kasutamine on lubatud ja isegi soovitatav.

Et aidata paremini mõista, kuidas ülesandele läheneda, on esile tõstetud mitu märksõna.

Erkspunasena esile tõstetud märksõnad on seotud andmetüüpide ja muutujatega.

**VARIANDID**

Teie programmi põhieesmärki täpsustatakse allpool. Ülesande variant vali oma sünnipäeva viimase numbri järgi, nt: 31.12.2001 on 1. variant.

0. Programmi põhieesmärk on lugeda täisarvude maatriksit, leida maatriksi iga rea ​​summa ja seejärel kuvada väljundis algmaatriks ja massiiv, mis sisaldab iga rea ​​summat.

1. Programmi põhieesmärk on lugeda täisarvude maatriksis, leida kõigi maatriksi põhidiagonaali all olevate elementide summa, seejärel kuvada väljundis algmaatriks ja kõigi põhidiagonaali all olevate elementide summa.

2. Programmi põhieesmärk on lugeda kahe mõõtmeline maatriks, leida maatriksi iga veeru jagatis ning seejärel kuvada väljundis algmaatriks ja massiiv, mis sisaldab iga veeru jagatist.

3. Programmi põhieesmärk on lugeda täisarvude maatriksit, leida maatriksi iga veeru korrutis ja seejärel kuvada väljundis algmaatriks ja massiiv, mis sisaldab iga veeru korrutisteid.

4. Programmi põhieesmärk on lugeda kahekordsete maatriksis, leida maatriksi iga rea ​​erinevus ja seejärel kuvada väljundis algmaatriks ja massiiv, mis sisaldab erinevust igast reast.

5. Programmi põhieesmärk on lugeda täisarvude maatriksist, leida põhi- ja kõrvaldiagonaali summa ning seejärel kuvada väljundis algmaatriks ja massiiv, mis sisaldab põhi- ja sekundaardiagonaalide summat.

6. Programmi põhieesmärk on lugeda ujuvpunktarvude maatriksis, leida maatriksis iga kõrvaldiagonaalist kõrgemal oleva positiivse elemendi korrutis ning seejärel kuvada väljundis algmaatriks ja iga diagonaali kohal olevate elementide korrutis.

7. Programmi põhieesmärk on lugeda reaalarvude maatriks, leida maatriksis negatiivsete elemendi summa ning seejärel kuvada väljundis algmaatriks ja kõigi negatiivsete elementide summa.

8. Programmi põhieesmärk on lugeda täisarvude maatriksis, leida maatriksi pea- ja kõrvaldiagonaali erinevus ning seejärel kuvada algmaatriks ja massiiv, mis sisaldab pea ja kõrvaldiagonaali erinevust.

9. Programmi põhieesmärk on lugeda täisarvude maatriksis, leida maatriksi esimese ja viimase rea jagatis ning seejärel kuvada algmaatriks ja massiiv, mis sisaldab esimese ja viimase rea jagatist.