Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Laboratorij za sisteme v realnem času

[VGS]: Vaje

Naloga 1 - Ponovitev C

Navodila

Navodila

Naloga je namenjena ponovitvi znanja iz programiranja v programskem jeziku C. Priporočeno je, da kodo pišete v razvojnem okolju, ki ga bomo tekom semestra uporabljali (STM32CubeIDE). Okolje podpira tudi uporabo razhroščevalnika, kar vam bo v prihodnjih nalogah v veliko pomoč. Na sistemu eŠtudij je naložen dokument z navodili za ustvarjanje projekta. Vsaka naloga spodaj naj bo v svojem projektu. Svetujem vam, da pri deklaraciji spremenljivk uporabljate datoteko *stdint.h* (link) (namesto *int* uporabite *int32_t*, namesto *short* uporabite *int16_t*, itd.).

Seznam nalog

1. Računske operacije in podprogrami

a. Vsota cifer identifikacijskega števil UM (IDUM)

Napiši program, ki bo izračunal vsoto cifer poljubnega števila, ki je zapisano (brez zunanjega vnosa) v programski kodi kot celoštevilčna spremenljivka (na primer $\mathrm{int\ a} = 5703$ ali $\mathrm{int\ b} = 123456$). V nalogi števila ne smete spremeniti v noben drug podatkovni tip.

b. Shranjevanje števil v polje

Napiši program, ki bo ustvaril polje 11 celih števil in v njega shranil prvih 11 naravnih števil, ki so deljiva hkrati z 3 in 4.

c. Drugo največje število

Napiši program, ki bo izmed 5 števil poiskal drugo največje.

d. Delitelji

Napiši program, ki bo med prvimi 100 naravnimi števil izpisal tiste, ki imajo natanko 3 delitelje. Zapišite funkcijo *ImaTriDelitelje*, ki kot vhodni parameter prejme celo število, kot rezultat pa vrne 1 (če ima natanko tri delitelje) ali 0 (če ima drugo število deliteljev). Funkcijo definirajte v .h datoteki, implementirajte pa jo v .c datoteki.

e. Zamik elementov v polju

Napiši program, ki bo ustvaril polje (1D) z velikostjo *M* in ga napolnil z naključnimi elementi. Program naj implementira naslednji postopek:

- Zamik elementov za eno mesto v desno.
- Element v zadnjem stolpcu se premakne v prvega.
- Program se zaključi takrat, ko je stanje polja enako kot na začetku.

Primer:

Dimenzija: m=5

Začetno polje:

1	3	5	7	9

Zamik polja:

1. korak
2. korak
3. korak
4. korak
5. korak

9	1	3	5	7
7	9	1	3	5
5	7	9	1	3
3	5	7	9	1
1	3	5	7	9

f. Pretvorba ure

Napišite program, ki bo število (predstavlja čas v desetinkah sekunde (1 = 100ms)) pretvoril v ure, minute, sekunde in desetinke sekunde. Število je zapisano v programski kodi (brez zunanjega vnosa). Pretvorjene vrednosti pretvorite v format: xxh yym zzs qqqms. xx v tem primeru pomeni, da bosta v izpisu ur vedno (vsaj) dve cifri. Za ustvarjanje niza si pomagajte s funkcijo *sprintf* (povezava), v pomoč pa vam bo še ta povezava.

Primer vnosa:	Izpis:
1	00h 00m 00s 100ms
15	00h 00m 01s 500ms
50000	01h 23m 20s 000ms

g. Iskanje podniza

Napiši funkcijo IskanjePodniza(char* niz, int dol), ki kot vhod prejme polje znakov in število znakov v polju. Funkcija naj v nizu poišče vse podnize, ki se začnejo z 'A' in se zaključijo z 'Z'. Če najde 'A' drugič, preden najde 'Z', se ga vzame kot del podniza. Funkcija izpiše vse najde podnize, vsakega v svoji vrsti.

2. Bitne operacije

a. Sodo ali liho število

Z uporabo bitnih operacij ugotovite:

- ali je število sodo ali liho,
- ali je število deljivo s štiri.

b. Sprememba predznaka

Napišite funkciji, ki bosta z uporabo dvojiškega komplementa podanem številu spremenili predznak. Prva metoda naj kot vhodni parameter sprejme 32-bitno celo število ter vrne enako število s spremenjenim predznakom. Druga metoda kot vhodni parameter sprejme naslov 32-bitnega števila in spremembo predznaka opravi na tej spremenljivki. Druga metoda ne vrača nič. Funkciji implementirajte v zunanji datoteki (npr. SpremeniPredznak(.c in .h)), ki jo dodate v glavni program, v katerem demonstrirate uporabo obeh funkcij.

c. Zlog po zlog z uporabo kazalcev

Napišite program, ki bo podano 32-bitno število razdelili na 4 zloge (uint8_t). Vsak zlog shranite v svojo spremenljivko tipa uint8_t. Za razdelitev uporabite kazalce.

d. Zlog po zlog z uporabo logičnih operacij

Napišite program, ki bo podano 32-bitno število razdelili na 4 zloge (uint8_t). Vsak zlog shranite v svojo spremenljivko tipa uint8_t. Za razdelitev uporabite logične operacije.

e. Števec bitov

Napišite funkciji, ki bosta za podano 16-bitno število prešteli, koliko bitov je postavljenih – imajo vrednost 1 (prva funkcija) in koliko je počiščenih – imajo vrednost 0 (druga funkcija). Dodajte tudi glavni program, v katerem boste pokazali pravilno delovanje napisanih metod. Funkciji ustvarite v ločeni datoteki (npr. StevecBitov(.c in .h)), ki jo dodate v glavno datoteko s stavkom *include*.

f. Postavi/počisti bit

Napišite funkciji, ki bosta določen bit (drugi parameter funkcij; predstavlja zaporedno vrednost bita, ki ga želimo spremeniti – skrajno desni ima vrednost 0) počistil oziroma postavil. Spremembo morate opraviti nad 16-bitnim število (prvi parameter funkcij). Funkciji ustvarite v drugi datoteki (npr. BitneOperacije(.c in .h)), ki jo dodate v glavno datoteko s stavkom include.

Primer klica funkcije:

PostaviBit(&stevilo, 0) – skrajno desni bit se postavi na vrednost 1,

PocistiBit(&stevilo, 2) – tretji bit od desne proti levi se postavi na vrednost 0.

g. Postavljanje dveh bitov

Napišite funkcijo, ki bo v 32-bitnem številu (prvi parameter metode) spremenil dva zaporedna bita (drugi parameter funkcije je indeks desnega – skrajno desni bit je 0) z vrednostjo, ki jo podamo (tretji parameter funkcije – ima lahko vrednosti 0 – 3 (2 bita)).

h. Iskanje vzorca

Napišite program, ki bo preštel, koliko krat se v 32 bitnem številu pojavi nek 3-bitni vzorec (npr. 101).

3. Strukture

a. Enosmerno povezan seznam opravil

Napišite strukturo Opravilo, ki vsebuje naslednje podatke:

- ime (niz znakov največje dolžine 20),
- prioriteta (celo število – 1 zlog),
- čas izvajanja (celo število – 1 zlog).

V glavnem programu ustvarite 3 opravila, ki jih uredite po prioriteti (od najmanjše do največje) in jih povežite v enosmerno povezan seznam z uporabo kazalcev.

Oddaja naloge

Pred oddajo vse datoteke s programsko kodo združite v eno datoteko formata zip.