

Nume și grupă:

## Introducerea în Organizarea Calculatoarelor și Limbaje de Asamblare

Model 2021-2022 (v1)

Timp de lucru: 120 de minute



1. Andreea este pasionată de comunicare secretă. Pentru aceasta pune la punct o schemă de comunicare cu prietena ei cea mai bună Bianca. Această schemă de comunicare folosește biții dintr-un mesaj trimis. Ajutați-o pe Andreea să preia codul secret dintr-un mesaj. Codul este obținut prin numărarea biților din mesaj. Folosiți scheletul din fișierul `num_bits.asm`. Recomandăm să folosiți o mască de biți cu un bit activat.

a. Pentru o valoare întreagă afișați dacă acea valoare are bit de semn sau nu, adică dacă este activ sau nu primul bit. Pentru testare folosiți variabilele `val1` și `val2` deja definite în program. (4 puncte)

b. Pentru o valoare întreagă afișați numărul de biți din acea valoare. Pentru testare folosiți variabilele `val1` și `val2` deja definite în program. (5 puncte)

c. Afișați numărul de biți pentru un mesaj efectiv, adică un vector de biți. Pentru testare folosiți variabilele `arr1` și `arr2` deja definite în program. (6 puncte)

Rezultatele așteptate pentru cele trei subpuncte pentru variabilele `val1` și `arr1`, respectiv `val2` și `arr2` se găsesc în fișierele `result1.txt`, respectiv `result2.txt`.

2. Dorin are un frate mai mic pe care dorește să-l învețe mai bine adunarea numerelor. Pentru că Dorin este pasionat de limbaje de asamblare, vrea să împuște doi iepuri dintr-o dată și să facă un program în limbaj de asamblare care să-l ajute pe fratele său să învețe adunarea numerelor. Ajutați-l pe Dorin să facă programul. Folosiți scheletul din fișierul `sum_parts.asm`.

a. Implementați funcția `sum` care calculează suma elementelor unui vector de întregi. Funcția primește ca argument vectorul de întregi și numărul de elemente ale vectorului. Apelul și afișarea funcției sunt deja realizate. (4 puncte)

b. Implementați funcția `sum_interval` care calculează suma elementelor unui vector de întregi între două poziții (`[pos1, pos2)`). Presupunem că pozițiile sunt valide. Funcția primește ca argument vectorul de întregi, prima poziție și doua poziție. Realizați și apelul și afișarea funcției. (6 puncte)

c. Folosiți funcția `scanf` pentru a citi cele două poziții de la utilizator și apoi apelați corespunzător funcția `sum_interval`. (5 puncte)

Rezultatele așteptate pentru exemple de rulări se găsesc în fișierul `results.txt`.

3. Monica a fost contactată de Serviciul Român de Informații Speciale Critice pentru a ajuta la obținerea unor mesaje capturate. SRISC i-a pus la dispoziție Monicăi un program de test (`strchr.asm`) pe care să își exerseze cunoștințele și încă unul care conține mesajele capturate, `marks.o`. Folosiți fișierele din arhivă pentru ca Monica să ducă la bun sfârșit sarcinile.

Pentru prima parte vedeți fișierul `strchr.asm`. Dezasamblați și urmăriți conținutul fișierului `marks.o` pentru a rezolva exercițiile. Veți scrie programul vostru în fișierul `test.c`. Folosiți-vă de fișierul `Makefile` și de fișierul `test.c` și executabilul `test` pentru investigația funcțiilor expuse de fișierul `marks.o`. Rezultatele așteptate se găsesc în fișierul `results.txt`.

a. Să se implementeze RECURSIV o funcție care calculează lungimea unui string asciz. Folosiți fișierul `strchr.asm` și comparați rezultatul cu funcția de bibliotecă `strlen`. (5 puncte)

b. Apelați funcția `first()` din fișierul `marks.o` pentru a afișa șirul `treasure`. (10 puncte)