

Nume și grupă:

Introducerea în Organizarea Calculatoarelor și Limbaje de Asamblare

8 februarie 2019 (v1)

Timp de lucru: 120 de minute



1. Costin a făcut primii pași în lumea avansată a numerelor naturale. Încercând să îl ajute, sora sa mai mare, Elena, pasionată de matematică și calculatoare, vrea să proiecteze un program care să-i verifice cunoștințele de sume și exponențiale. Ajutați-o pe Elena să implementeze acest program pornind de la fișierul `sum_n_alt.asm`. Scopul final este să calculați suma alternativă a puterilor unui număr natural, adică $3^0 - 3^1 + 3^2 - 3^3 \dots$

a. Realizați puterea unui număr natural și afișați-o. Folosiți `base` pentru numărul natural (*baza*), `len` pentru putere (*exponentul*). (4 puncte)

b. Realizați suma alternativă a primelor numere naturale și afișați-o. Adică $0 - 1 + 2 - 3 + 4 \dots$. Folosiți `len` pentru numărul de numere naturale pentru care calculați suma alternativă. (5 puncte)

c. Realizați suma alternativă a puterilor unui număr natural adică $3^0 - 3^1 + 3^2 - 3^3 \dots$. Folosiți `base` pentru numărul natural (*baza*), `len` pentru putere (*exponentul*). Puteți calcula suma în variabila `sum` sau într-un registru, cum doriți. (6 puncte)

Rezultatele așteptate se găsesc în fișierul `results.txt`. Modificați corespunzător valoarea constantei `len` pentru verificare.

2. În urmă cu puțin timp, Adriana a descoperit geometria. Adriana este pasionată de patrulatere și dreptunghiuri și vrea să facă multe calcule cu aceste figuri. Ajutați-o pe Adriana să implementeze calculul perimetrului și ariei unui dreptunghi pornind de la fișierul `rectangle.asm`. În fișier `first_array` este un vector cu lățimi, `second_array` este un vector cu lungimi.

a. Implementați funcția `perimeter()` și apelați funcția pentru a calcula perimetrul dreptunghiului dat de lățimea `first_array[0]` și lungimea `second_array[0]`. (4 puncte)

b. Implementați funcția `area()` și apelați funcția pentru a calcula aria dreptunghiului dat de lățimea `first_array[0]` și lungimea `second_array[0]`. (4 puncte)

c. Calculați, respectiv, în `perimeter_array` și `area_array`, perimetrul și aria dreptunghiurilor cu lățimile și lungimile indicate de `first_array` și `second_array`. Afișați vectorul `perimeter_array` și `area_array` folosind funcția `print_array()`. (7 puncte)

Rezultatele așteptate se găsesc în fișierul `results.txt`.

3. Luca a fost mereu pasionat de puzzle-uri, jocuri logice și escape rooms. De când este student la facultate, a dezvoltat un interes în inginerie inversă. Să se prindă cum funcționează un program sau un dispozitiv urmărind funcționalitatea și investigând internele. Executabilele și limbajul de asamblare sunt cea mai recentă pasiune. Acum a pus mâna pe niște programe care îi vor testa cunoștințele. Ajutați-l să rezolve provocările.

Pentru prima parte vedeți fișierul `memset.asm`. Pentru celelalte două părți, dezamblați și urmăriți conținutul fișierului `escape.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 `escape.o`. Rezultatele așteptate se găsesc în fișierul `results.txt`.

a. Actualizați fișierul `memset.asm` pentru a umple șirul `store_str` cu 'A' (caracter) folosind operații pe șiruri, ca alternativă la apelul funcției `memset()`. (5 puncte)

b. Apelați funcția `box()` din fișierul `escape.o` pentru a afișa șirul *think outside the box*. (5 puncte)

c. Apelați funcția `cube()` din fișierul `escape.o` pentru a întoarce numărul 4727. (5 puncte)