

Nume și grupă:

## Introducerea în Organizarea Calculatoarelor și Limbaje de Asamblare

14 ianuarie 2019

Timp de lucru: 90 de minute



1. Marian este cercetător spațial la NASA. Munca sa presupune calcularea unor distanțe dintre diferite planete, sisteme solare, galaxii. Distanțele aceasta sunt atât de mari încât nici cele mai performante calculatoare pe 32 de biți de la NASA nu au mecanisme *built-in* de a însuma aceste numere. Din acest motiv, Marian te-a contactat pe LinkedIn pentru a îl ajuta să implementeze această facilități. În schimbul acestui serviciu, îți va pune o vorbă bună la NASA. Implementați pornind de la fișierul `thule.asm`.

a. Implementați adunarea a 2 numere pe 4 octeți. Folosiți variabilele `word_num1`, `word_num2` și `result_word`. (2 puncte)

b. Implementați adunarea a 2 numere pe 12 octeți. Folosiți variabilele `big_num1`, `big_num2` și `result_4word`. Atenție la *carry*! (5 puncte)

c. Implementați adunarea a 2 vectori de numere pe 12 octeți. Folosiți variabilele `num_array1`, `num_array2`, `result_array`. Atenție la *carry*! (8 puncte)

Pentru fiecare pas afișați rezultatul (ideal în hexazecimal) pentru verificare.

2. Nicolae este foarte pasionat de numerele prime. De fiecare dată când îi vine în minte un număr, trebuie să calculeze să vadă dacă este prim, altfel nu se poate concentra la nimic altceva. Dându-și seama că a face calculele pe hârtie durează prea mult, iar viața este prea scurtă pentru a descoperi toate numerele prime, acesta se decide să implementeze un program pe calculatorul vechi al tatălui său, care să îi calculeze instant această proprietate. Problema este că Nicolae nu este un programator prea bun, iar calculatorul tatălui sau este atât de vechi încât nu știe să ruleze decât programe scrise în limbaj de asamblare pe 32 biți. Ajutați-l pe Nicolae să își scrie programul pentru a se putea bucura de numerele prime. Implementați pornind de la fișierul `goldbach.asm`.

a. Folosind funcția `fgets()`, citiți un număr introdus de la tastatură. (3 puncte)

b. Funcția `unsigned int stringToNumber(char *str)` primește un pointer către un șir de caractere și întoarce numărul reprezentat de șirul `str`. Implementați funcția `stringToNumber()`. Șirul conține doar cifre; nu trebuie să tratați cazurile în care șirul conține alte caractere. (6 puncte)

c. Nicolae a implementat funcția `isPrime()`; din păcate acesta nu este familiarizat cu convenția de apel specifică procesoarelor Intel pe 32 de biți. Corectați greșelile/omisiunile lui Nicolae. (6 puncte)

3. Cei de la NSA au interceptat niște binare despre care se bănuiește că ar avea legătură cu portofelul electronic al lui Snowden. Aceștia însă nu au reușit descifreze și să spargă binarele respective așa ca ți-au cerut ajutorul. Ajută-i să înțeleaga mai bine binarul.

a. Navigați în directorul `1-buffer-overflow/`. Analizați binarul. Găsiți inputul pentru care funcția `getWallet()` va fi apelată. În cazul în care ați reușit, va fi afișat mesajul *Receiving wallet!*. (5 puncte)

b. Navigați în directorul `2-buffer-overflow/`. Analizați binarul. În funcția `transferMoney()` se realizează un apel către `printf()` în anumite condiții. Găsiți inputul pentru care se apelează funcția `printf()`. În caz de succes, programul va afișa mesajul *Transferring Money!*. (5 puncte)

c. Reveniți în directorul `1-buffer-overflow/`. Găsiți inputul pentru care programul se încheie cu succes (fără *segmentation fault*). (5 puncte)