Documentatie tema ASC

Bouruc Petru-Liviu, 135, tema de 10

Programul incepe cu algoritmul de determinare a unui numar prim. Astfel, el trateaza cazurile in care este mai mic decat 2, cand este 2, cand este divizibil cu 2 si in final verifica fiecare numar impar pana la radical din p daca este divizor al lui p. Daca numarul nu este prim, atunci algoritmul afiseaza mesajul corespunzator si se incheie rularea.

Daca numarul este prim, atunci verificam in mod repetitiv fiecare numar mai mare decat 2 daca este generator. Un numar x este generator daca si numai daca ridicat la puterea p-1 da 1 si nu exista o alta putere *a* mai mica decat p-1 astfel incat x^a = 1 (mod p).

Pe parcurs, puterile lui x sunt salvate in vectorul v.

Pentru codare si decodare, se ia fiecare litera din mesaj si se scade 65 din codul ASCII al caracterului pentru a afla pe ce pozitie *poz* se afla in alfabet.

Astfel, mesajul codat va fi format din litere aflate pe pozitiile k = v[poz] pentru fiecare caracter c din sirul initial, afisandu-se alfabet[k]. Codarea se incheie atunci cand caracterul curent este ‘\n’ care are codul 10 in ASCII.

Pentru decodare, se cauta in vector printr-o parcurgere pe ce pozitie *j* se afla *poz* in vector, urmand ca la final programul sa afiseze alfabet[j].

**Codul:**

.data

v:.space 400

sir1:.space 400

sir2:.space 400

p:.space 4

nl:.byte '\n'

alfabet:.asciiz "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

afisNPrim:.asciiz "Numarul nu este prim"

gen:.asciiz "Generatorul g este: "

.text

newline:

lb $a0, nl

li $v0, 11

syscall

jr $ra

main:

li $v0, 5

syscall

move $t0, $v0

sw $t0, p #citesc valoarea p si o salvez

#verificam acum daca p este prim

blt $t0, 2, nuPrim #daca este mai mic decat 2 atunci nu este prim

beq $t0, 2, ENDloopPrime #daca este 2 atunci sarim peste restul procesului de verificare

rem $t1, $t0, 2

beq $t1, 0, nuPrim #daca este divizibil cu 2 nu este prim

li $t1, 3 # $t1 este indicele i din for cu care verificam daca este un divizor al lui p

loopPrime:

mul $t2, $t1, $t1 #verific daca i \* i > p

bgt $t2, $t0, ENDloopPrime

rem $t3, $t0, $t1

beq $t3, 0, nuPrim #daca l-am gasit pe $t1 ca divizor, atunci nu este prim

addi $t1, $t1, 2 #mergem din 2 in 2, doar pe numerele impare

j loopPrime

nuPrim:

la $a0, afisNPrim

li $v0, 4

syscall

j exit

ENDloopPrime: # acum cautam generatorul g

li $t1, 1 # cautam g; $t1 = g

li $t7, 0 # t7 este indexul in vector; am pus t7 deoarece am modificat codul mai tarziu

sw $t1, v($t7) # g^0 = 1

addi $t7, 4

beq $t0, 2, gasitG # daca p = 2 atunci singurul generator este 1

cautaG:

li $t7, 4 # cand cautam un nou generator, indexul se intoarce pe pozitia 2

addi $t1, $t1, 1 # incepem cu 2 deoarece 1 nu poate fi generator

sw $t1, v($t7)

addi $t7, 4

move $t2, $t1 # $t2 = g\_anterior

li $t9, 1 # $t9 reprezinta puterea la care se ridiga g

# am ales t9 deoarece este mai usor de urmarit puterea la care se ridica g

# singura posibilitate ca g sa fie generator este daca ridicat la p-1 sa dea 1, si sa nu mai

# existe o putere mai mica b, exceptand 0, astfel incat g^b = 1

subu $t8, $t0, 1 # mergem pana la p-1 = $t8

ridicaPut:

addi $t9, 1 # incepem verificarea mai intai cu g^2

mul $t2, $t2, $t1 # inmultim g\_ant cu g

rem $t2, $t2, $t0 # facem (g\_ant\*g)%p

sw $t2, v($t7)

addi $t7, 4

beq $t2, 1, verif1 # verificam daca g^b = 1 si b = p-1

j ridicaPut

verif1:

beq $t9, $t8, gasitG

blt $t9, $t8, cautaG

gasitG: #am gasit generatorul, acesta fiind salvat in $t1

la $a0, gen

li $v0, 4

syscall

move $a0, $t1

li $v0, 1

syscall

jal newline # in momentul de fata avem $t0 = p si $t1 = g

subu $t7, $t7, 4

sw $0, v($t7)

la $a0, sir1

li $a1, 400

li $v0, 8

syscall

li $t9, 0 # $t9 este indexul in sir1 (primul sir citit)

codeazaV:

lb $t2, sir1($t9) # in $t2 am caracterul curent

addi $t9, 1

li $t5, 10 # 10 este codul ascii pt newline

beq $t2, $t5, sfCodeaza

subu $t2, $t2, 65 # acum trebuie sa-l gasim pe g^t2 si sa-l afisam

mul $t2, $t2, 4

lw $t3, v($t2)

lb $a0, alfabet($t3)

li $v0, 11

syscall

j codeazaV

sfCodeaza:

jal newline

la $a0, sir2

li $a1, 400

li $v0, 8

syscall

li $t9, 0

decodeaza:

lb $t2, sir2($t9)

addi $t9, 1

beqz $t2, exit

subu $t2, $t2, 65

li $t7, 0

li $t8, 0

cautaVal: # cautam pe ce poz se afla t2

lw $t3, v($t8)

beq $t3, $t2, aflatPoz

addi $t7, 1

addi $t8, 4

j cautaVal

aflatPoz: # l-am gasit la poz $t7

lb $a0, alfabet($t7)

li $v0, 11

syscall

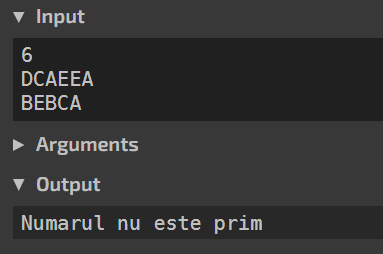
j decodeaza

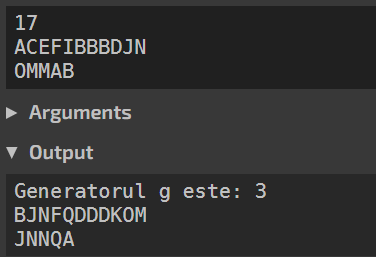
exit:

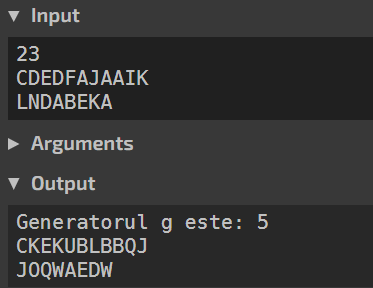
li $v0, 10

syscall

**Exemple:**







Link tio.run : https://tio.run/?fbclid=IwAR31Dipv\_00TN-dWrMaG2YTDxnq27IL7aqD9ey3bLNJ61ULYOyTboOcbWvw##fVdpb9w2EP0s/grCcRE7WBvadVPbClzAiRM3R133cg8ULrgSd8FAVyRqk@yfd98MSR0Lo0ZsSNRwOPPezBumrU3x8HCcKatEtEmO21qlWn4bxyJqTTPfXVhMFur@TURlnhwvv1otn/5TPhWRyldqqW1yrNrUmK3cu3z56ur1m@sf3r57/@HHm59uf/7l199@v/vjz7/@3oP1yrQ3t40pBvubrlBNl8uyk7qF2xpfYbnW5WBzrUvdKFuR3ZrNErknjq3@YkWpP@em1ImI8qXcV/FMljmejdzf4Hk@Rzpf21TlWPzYyP1GCVEoUyaDzfORSVFttNy3WMUnrH92L7Uc/UT4eZIaq9tUblReqUYrWLRGVrJV@UZvRSSiJxvdmJVJVSFV2hUyUylZ9TmKaJlb532BmDtCZXDP1myLYGVhUpnpVFm5kMp2ZWomcMGV/tS7en1zlVdVTf701FW/uVV0WM2LDf4C1rqpUt12eWdwkgyxN1pEjS7gez7zB4TDsBCHuHdCzszGbM3S5DLtcOgkVAc77T6JeBu9uO@mzEyqczzAQylXVUP7KQg5YDmc0pXuIJipXFLgNZyHzMFvVCCvfbuYhfDndJ535RwZ@Qy/39POaLm2wTqeooiP@OdwOPHf4Y32MBQnUygcEvkRol2r1lgAzUli0Qc8e5RF@FNZZvpoATZ5K3SzxtEEyULSn5nMUHPktezwkRErauYq@iiHqNGrHBJBkSvXG30D0pqvfzR13wDsQn8xVuBxjEHi2eJaTlVnKbtxWw7Ezr1psHrB2V/0JqeAy5vY0556/QVekN5Gp/D4QmJn3bVkkWlqMagPlooq40KwMq0ybKDusKrZms43K87fHOCMQ3/C@j7G0eDKQ3vK@Y77hTm6dta@SS9GrWLKdccJhmRdxPDI@V0nHkn2HBIvs5A9qrSsumH3rM@1pbQtUkNm4LKutsYaRQ22WwZctzBOdY1C4J7qMZlTDdWVQkgrMxxDlE7gGLw6BKg9nNYtQmtwLy6IqH9VadEm7IZzO/esksU5JKNu9BbBA6sOhpA/1Bc3KpJqTGbWitiOQs1AAnMNLs@HuHtt61pACsnpGswBapbHPaYTj44VHF@1pDPGUv5kAmkDDnqXLebVubHkuT6ak2WGc4BQS4pIOFI8wyFoA9pb@ZCCEiu5nOEbyLDEMzVVa1eaipe8r@@XruCitlt2wOvMK4Zj0XdzrUoVArkgIxIYju@2s1RSga3zXfoHaVYcEdFgqCjW91Q6Y9ELzNJmLFtoPlPL1mTrJG2wjcl2hUFfyAO2fLY@/Ia00RXTYlRMk2qid9dSMKGSoxj54B3h9uAQ4vSA7Gkr/X6UffpObp2PJBp8n88cmq5heZ0naFh3/Rj4Q1DOMPF89nI8ki1wh6lnqWbQljy9QSFUiOU9iCY2iLFa7twWVBgH/aVjfOfAbPJXFF9W8F5UhS5p6qLuV7iRSbVhJkiq@CIRBFOEIjqd9Z1LVMQDE0OYdI9zQShwgItbNIR0NgpJ9D0dR0NT7@oweZMHNJhIqwxmMeTJ4kAIr1ZbdcfCt3SckzHiOT8MGZKQKBoUjUrRPB1dBRrkLCaF7ZjmaJ7jNXbRzGMXjFN4vgPK2gYUxajUeFu7euUict48YKGov3s@jC3b6GUHcWjVUc61QHcg1CRiZQnAKs1HVYjdNqIRmX92o56AXxz65Hmmuiswlk8OR2N1Pt@dqwE5oqCPOpnUyA6fiwmfj9I5YlNEuCT2XsfkLJicHfAdkltnxDP/cQDFaMLF4fnMPXPP3ak8mYx8oJryRCP1ViskZFmbxhCeHQ6dfeLPIlN7W23FVGDmo9czRwah6Y92UYetPpDR7YtUtqIsTx/l7PT/OesRFSwpBFMy3hAPGx4eFicC//m5enP57vLy7Xvx4ebq8uXr95f/AQ