Arhitectura Sistemelor de Calcul – Tema

*Autor: Amzuloiu Sergiu – Grupa C112E*

# *Cuprins*

[TODO1: SEED **2**](#_Toc136020232)

[**Sumar** 2](#_Toc136020233)

[Rezolvare 2](#_Toc136020234)

[TODO2: RAND **5**](#_Toc136020235)

[Sumar 5](#_Toc136020236)

[Rezolvare 5](#_Toc136020237)

[TODO3: ENCRYPT **6**](#_Toc136020238)

[Sumar 6](#_Toc136020239)

[Rezolvare 6](#_Toc136020240)

[TODO4: ENCODE **7**](#_Toc136020241)

[Sumar 7](#_Toc136020242)

[Rezolvare 7](#_Toc136020243)

# TODO1: SEED

## Sumar

Sa se calculeze valoarea termenului initial din cheia de criptare.

## Rezolvare

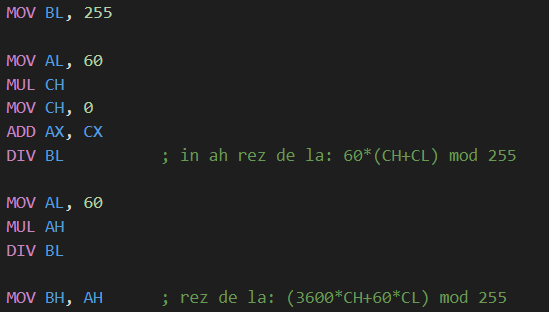
Termenul initial al sirului, x0, a fost calculat cu ajutorul functiei :

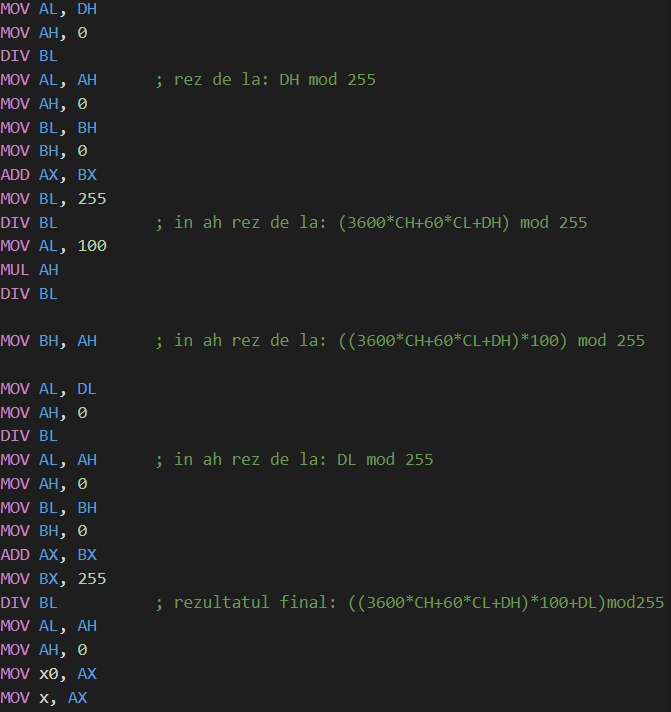
x0 = ((CH ∗ 3600 + CL ∗ 60 + DH) ∗ 100 + DL) mod 255,

unde registrele CX si DX sunt folosite pentru a stoca ora actuala a sistemului prin intermediul vectorului de intrerupere 21h, prin subfunctia 2Ch.



Pentru calculul termenului, a fost distribuit modulo 255 in vederea simplificarii urmaririi si a utilizarii registrelor (rezultatul este echivalent cu utilizarea operatiilor pentru operanzi de tip word).





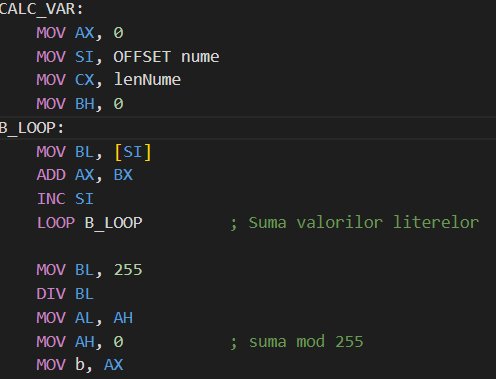
In aceasta subrutina, a fost apelata subrutina CALC\_VAR prin intermediul careia au fost calculati coeficientii a si b.



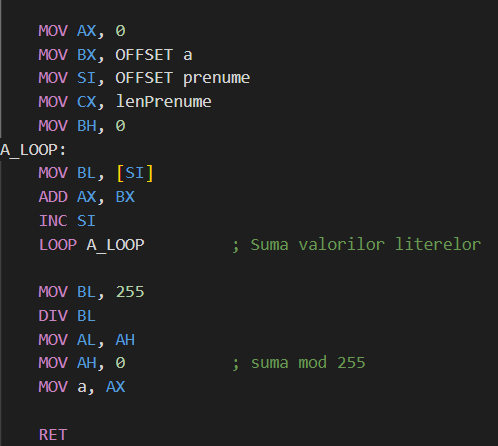
Coeficientului a i-a fost atribuita suma valorilor prenumelui mod 255, iar coeficientului b i-a fost atribuita suma valorilor numelui mod 255.

Valorile au fost adunate intr-un registru intermediar, apoi a fost efectuata operatia de div.

Pentru b :



Pentru a :



# TODO2: RAND

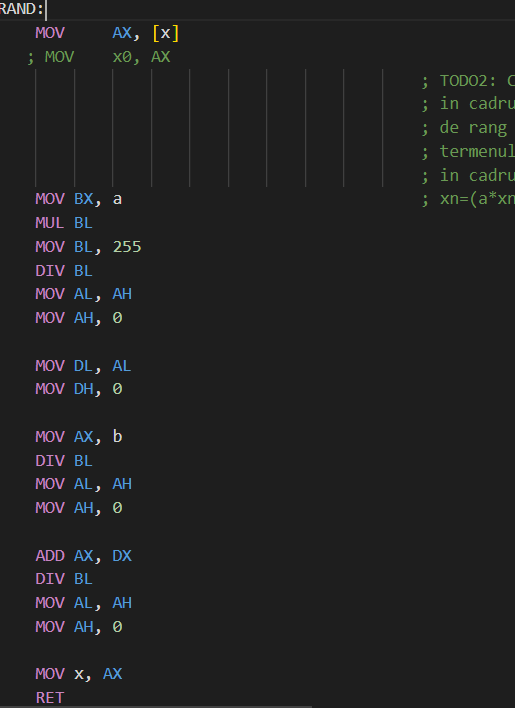
## Sumar

Sa se completeze subrutina RAND pentru a calcula termenul de rang n.

## Rezolvare

Termenul de rang n a fost calculat pe baza coeficientilor a, b si a termenului de rang inferior (n-1), folosind formula :

xn = (a ∗ xn−1 + b) mod 255



De asemena, a fost distribuit modulo pentru fiecare termen al operatiei.

# TODO3: ENCRYPT

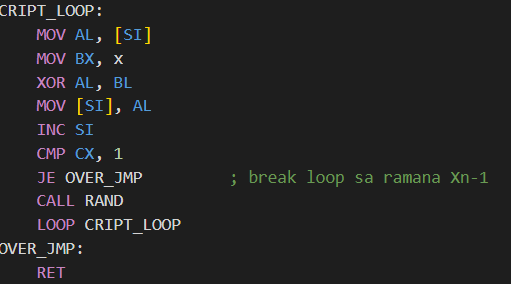
## Sumar

Sa se realizeze criptarea mesajului din fisierul de intrare.

## Rezolvare

Criptarea este sub forma unei operatii de XOR intre fiecare octet din sirul de intrare si xn-ul corespunzator pozitiei octetului.

Criptarea a fost realizata prin intermediul unui loop in care a fost generat termenul xn si efectuata operatia de xor dintre acesta si mn, unde mn este octetul cu pozitia m din mesajul de intrare.



In urma criptarii, xn-1 a fost generat si pastrat pentru afisare (prin comparatia lui cx cu 1), iar mesajul criptat va fi folosit pentru codare.

# TODO4: ENCODE

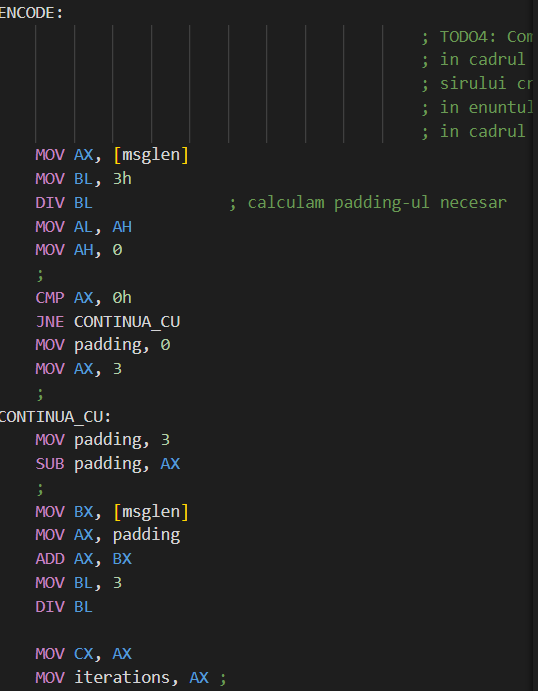
## Sumar

Sa se realizeze operatia de codificare asupra sirului criptat.

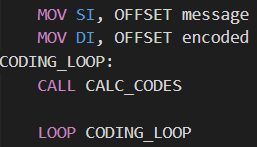
## Rezolvare

Codificarea COD64 a fost realizata prin gruparea a cate 3 octeti in grupari de cate 4 a 6 biti, urmand ca fiecare 6 biti sa fie indexati.

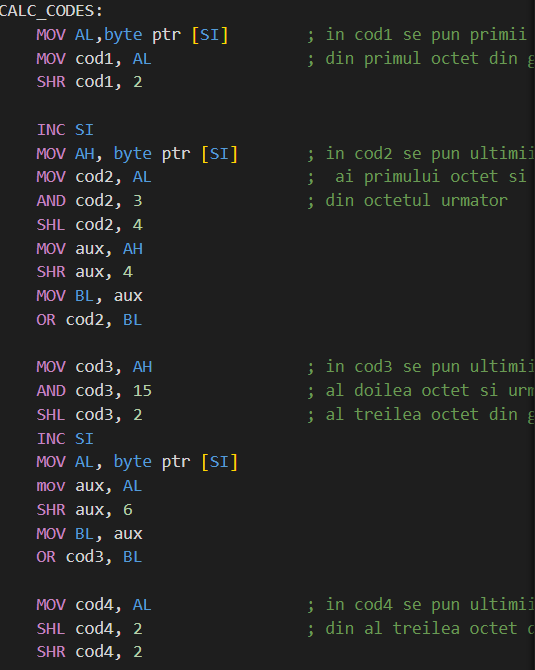
Initial, a fost verificat daca lungimea sirului de intrare este divizibila cu 3, urmand sa se seteze padding-ul necesar. In cazul in care restul impartirii lungimii la 3 este 1 padding-ul este de 2 octeti, iar daca restul este egal cu 2, padding-ul este de un 1 octet.



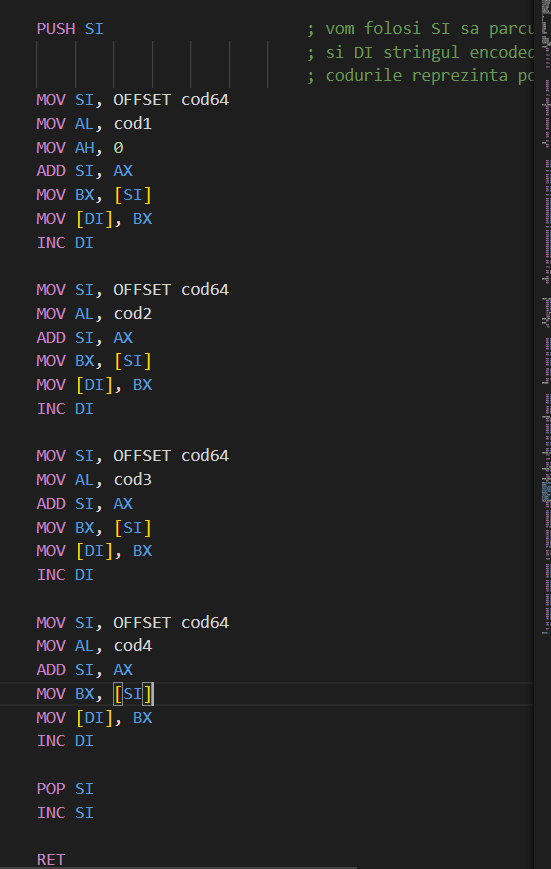
In secventa urmatoare de cod este scris un loop pentru codare in care vom lua fiecare grupare de cate 3 octeti din sirul initial si o vom translata in cate o grupare de 4, fiecare octet avand pe ultimii 6 biti valorile corespunzatoare.



In subrutina CALC\_CODES au fost calculati cei 4 octeti cu ajutorul variabilelor cod1, cod2, cod3, cod4 si aux si prin intermediul shiftarilor si folosirii mastilor de biti.

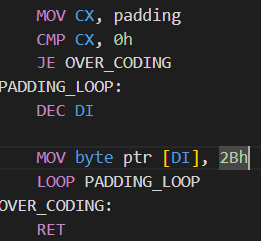


In aceeasi subrutina, sirul de caractere encoded iterat prin DI, a fost populat cu valorile corespondente din baza de codare.



Offset-ul bazei de codare a fost salvat in SI, iar valorile pentru codificare au fost aduse din memorie cu ajutorul variabilelor cod1, cod2, cod3, cod4 reprezentand un offset pentru acestea.

Dupa codificare, au fost adaugate valorile pentru octetii de padding in sirul encoded.



Pentru usurinta intelegerii, este atasata secventa de cod scrisa in limbajul C pentru aceasta codare :

