

Echipa DBB- Persa Diana, Nodis Bianca, Mranovatz Brigitte

Programul citeste de la tastura un sir de 8-16 octeti, in format hexa. Octetii sunt convertiti in binar si salvati intr-un sir. Pe baza sirului , se construieste un cuvant C, calculat prin bitii octetilor . Sirul este sortat descrescator , se calculeaza octetul cu numar maxim de biti 1 si pozitia din sirul sortat este afisata in baza 10. Mai departe, pentru fiecare octet din sir se calculeaza suma primilor doi biti N. Fiecare octet este rotit spre stanga de N ori. Noul sir este afisat in baza 2 respectiv 16.

In *main.asm*, se ruleaza programul principal unde se citeste sirul de la tastura, procesat apoi manipulat .

In fisierul *diana.asm* se gasesc procedurile de calculare a cuvantului C, construirea unui sir de octeti rotiti si afisarea acestora in baza 2 si 16 .

Fisierul *brigitte.asm* contine sortarea sirului de octeti , calcularea maximului de biti 1 a unui octet , cu precizarea ca acest maxim este mai mare ca trei. Afisarea sirului sortat in baza 10 dar si pozitia in sir al octetului cu numar maxim de biti 1.

Citire

Citirea se realizeaza folosind functia INT 21h, AH=0Ah, care preia un sir de caractere de la tastatura si il stocheaza intr-un buffer de memorie. Programul asteapta de la utilizator o secventa de caractere reprezentand valori hexazecimale, separate prin spatii.

Convertire din hexa in binar

Conversia transforma sirul de caractere ASCII introdus de utilizator in valori numerice binare. Mai intai, programul extrage prima cifra si o converteste intr-o valoare numerica intre 0 si 15, apoi o deplaseaza pe pozitia nibble-ului(4 biti) superior prin mutarea bitilor la stanga. Aceasta valoare este combinata printr-o operatie logica de tip OR cu a doua cifra convertita, rezultand astfel un octet complet. In timpul acestei operatii, programul ignora automat spatiile goale dintre caractere si contorizeaza numarul total de octeti generati pentru a stabili dimensiunea sirului ce va fi trimis ulterior catre procedurile de sortare si calcul.

Construirea cuvantului C

Cuvantul C este construit pe baza valorilor octetilor din sir . Pentru bitii 0-3 (ai lui C) se aplica XOR intre primii 4 biti al primului octet si ultimii 4 biti al ultimului octet. La bitii 4-7 se realizeaza OR intre toti bitii 2-5 al fiecarui octet din sir , rezultatul final se reduce la 4 biti. Bitii 8-15 reprezinta suma tuturor octetilor modulo 256. Deci , restul sumei a tuturor octetilor impartita la 256 este pusa in bitii 8-15 . Acest numar este afisat in hexa cat si in binar.

Sortarea descrescatoare a sirului

Algoritmul de sortare folosit este bubble sort. Sirul este parcurs prin doua loop-uri, important de mentionat ca fiecare loop este realizat de n-1 ori . Fiecare elementele sunt comparate cu vecinii

sai, iar dupa caz, interschimbat. In asa mod , pentru fiecare iteratie , unul dintre elemente se afla pe pozitia corecta pe care o ocupa in sirul sortat.

Calcularea octetului cu numar maxim de biti 1

Fiecare octet din sir este parcurs. Pentru fiecare octet se efectueaza 8 rotiri spre stanga folosind Carry Flag (CF) .Se verifica valoarea din CF , daca este unu se contorizeaza. Dupa cele 8 rotiri a unui octet, se compara elementul cu 3, deoarece problema impune ca $\text{maxim} > 3$. Daca trece aceasta conditie, contorul este comparat cu maximul curent. In cazul in care contorul este mai mare decat maxim, acesta se actualizeaza.

Afisarea in baza 10

Pentru afisarea unei valori pozitive in baza 10, se executa impartiri succesive la 10, iar resturile obtinute sunt salvate intr-o stiva. Ulterior, prin respectarea principiului LIFO (Last In, First Out), oferit de stiva ,cifrele sunt extrase si afisate, rezultand astfel reprezentarea zecimala corecta a numarului.

Construirea sirului de octeti rotiti

Sirul este construit pe baza valorilor din sirul citit de la tastatura. Pentru fiecare element, se determina numarul de pozitii de rotire N, calculat ca suma primilor doi biti (bitul 0 si bitul 1) ai octetului curent.Apoi, octetul este rotit spre stanga cu N pozitii folosind instructia ROL, iar rezultatul este memorat intr-un sir nou. Procedura utilizeaza contorul CX pentru parcurgerea sirului si restaureaza registrele folosite la final.

Afisarea in baza 2

Pentru afisarea unui octet in baza 2, se foloseste deplasarea spre stanga cu Carry Flag. Se face un loop de 8 ori , unde se verifica fiecare bit al unui octet, prin intermediul lui CF. Daca $\text{CF}=1$ se face salt la o eticheta care afiseaza caracterul '1', altfel '0'. Afisarea se face prin intreruperea 02h a functiei int 21h.

Afisarea in baza 16

In cadrul afisarii in baza 16, este nevoie de o tabela care contine elementele acestei baze. In binar, fiecare grup de 4 biti are o prezentare hexa. Se izoleaza 4 biti din octet iar apoi se foloseste instructiunea xlat. Aceasta instructiune are ca scop obtinerea caracterului corespunzator din tabela oferita de utilizator. In cazul acesta, baza 16 . Aceasta tabela se numeste tabela de translatare. In asa mod , din baza 2 se ajunge la baza 16. Acest proces se repeta pt fiecare octet de doua ori.

Probleme intalnite:

O problema intalnita a fost la sortarea sirului . Loop-uri se executau de n ori , iar compilatorul accesa garbage values.Problema a fost rezolvata simplu , doar decrementand lungimea cu 1.

O alta problema a fost la unele proceduri atunci cand foloseam mai multe loop-uri. Instructiunea loop foloseste registrul CX, iar fiecare loop il modifica. Ca sa nu pierdem valoarea lui intre

loop-uri sau apeluri de proceduri, a trebuit sa salvam CX pe stiva inainte si sa il recuperam dupa.