Definiti notiunea de tip de data la nivelul limbajului de asamblare. Care sunt mecanismele de definire ale unui tip de data la nivelul limbajului de asamblare? Dati exemple de declarare de variabile (cu si fara initializare) pentru toate tipurile de date. Care sunt situatiile in care operatorii de tip ai operanzilor sunt necesari si cand nu? Explificati si justificati prin scurte exemple adecvate si suggestive de cod sursa fiecare dintre cazurile prezentate. Efectuati o clasificare a conversilor de tip (przentand, exemplificand prin scurte secvente de cod sursa si comentand modalitatiile de efectuare a conversilor de tip la nivelul limabjului de asamblare 80x86.

ELEMENTE DE BAZA ALE LIMBAJULUI DE ASAMBLARE

La nivelul limbajului de asamblare, notiunea de **tip de data** face referire la modul de reprezentarea si manipulare a datelor intr-un program.

Definirea unui tip de data se realizeaza cu ajutorul **directivelor de alocare a spatiului** (DB, DW, DD, DQ), ce sunt utilizate pentru a specficia spatiul de reprezentare a datelor, generand in memorie numarul de octeti specificati (exemplu: nume\_variabila DB 10 va genera un octet / byte in memorie pentru variabila respectiva, iar variabilei ii va atribui valoarea 10)

*Exemple declarare cu initializare: Exemple declarare fara initializare:*

a db 3 a db

b dw 1,2,3 b dw

c dd -1 c dd

d dq 123456h d dq

Operatorii indica modul de combinare a operanzilor in scopul formarii expresiei. Mai exact, **operatorii de tip** specifica tipul unor expresii sau a unor operanzi pastrati in memorie, iar sintaxa pentru acestia este *tip expresie*, unde specificatorul de tip este unul dintre cuvintele cheie BYTE, WORD, DWORD, QWORD. Acesti specificatori au intotdeauna rolul de a clarifica o ambiguitate (inclusiv cand este vorba despre o variabila de memorie).

Specificatorul de dimensiune al operanzilor este **necesar** in urmatoarele cazuri ce folosesc variabile de memorie / adrese :

* Inmultire si impartire: (i)mul byte[b] / (i)div dword[a]
* In cadrul instructiunii mov: mov word[a], 10
* Cu lucrul pe stiva: push dword[n] / pop dword[n] (! Pt push 15 nu se va emite eroare si asamblorul va face push dword 15)
* In cadrul lator instructiuni precum: inc dword[n] / dec dword[n]

**Conversiile de tip** – se refera la extinderea reprezentarii unei date prin completarea anterioara cu bitul de semn (in cazul conversiilor cu semn) sau zerorizare (in cazul ocnversiilor fara semn) - pot fi incadrate in 3 mari categorii:

1. In functie de natura lor
   1. Distructive: cbw/cwd/cdq/ mov ah, 0/mov edx, 0/movzx d, s/movsx d, s (are loc suprascrierea)
   2. Nedistructive: operatorii de tip: byte, word, dword, qword
2. In functie de semn
   1. Cu semn: cbw/cwd/cdq/movsx d, s
   2. Fara semn: movzx d, s/mov ah, 0/mov edx, 0/operatorii de tip
3. In functi de modul prin care se realizeaza
   1. Prin largire (fara pierdere de informatie) : toate cele distructive + word/dword/qword
   2. Prin ingustare: byte/word/dword