

# Curs 6

## Aritmetica de pointeri, L-value, R-value:

- adunarea a doi pointeri (OPERATIE INTERZISA ÎN INFORMATICĂ)
- Exemple de operații între pointeri:
  - o scăderea a doi pointeri (exemplu: determinarea lungimii unui sir) – rezultat = constantă numerică (valoare imediată) = scalar
  - o adunarea unei constante la un pointer (exemplu: localizarea unui element din memorie) – rezultat = pointer
  - o scăderea unei constante la un pointer(exemplu: localizarea unui element din memorie) – rezultat = pointer
- `mov ax, a[7]`  
`mov ax, 7[a]`  
`mov ax, [a + 7]`
- $a[7] = *(a + 7)$
- $i = a[7]$   
 $i = *(a + 7)$   
 $i = *(7 + a)$   
 $i = 7[a]$
- $i = i + 1$   
 $i$  = adresă, LHS, L-value  
 $i + 1$  = conținut, RHS, R-value
- $<\text{identificator}> = <\text{expresie}>$
- $<\text{expresie\_calcul\_de\_adresă}> = <\text{expresie\_aritmetică}>$   
`mov [ebx + edx * 5 + v - 7], a + 2`  
 $(a + 2 ? b : c) = x + y + z$       CORECT  
 $(a + 2 ? 1 : c) = x + y + z$       SYNTAX ERROR
- C++ reference variables (&):
  - 1) `int & j = i;` // j devine alias pentru i
  - 2) transmitere de variabile prin referință:  
`float f(int &x, int y)`
  - 3) returnarea de L valori prin intermediul funcțiilor  
`int v[200];`  
`int& f(x, i) { ... return v[i]; }`  
 $F(a, 79) = 141493; //v[79] = 14193$   
 $q = f(a, 79)$  sau  $q = v[79]$

- mov eax, ebx – corect
- mov eax, [ebx] – corect
- mov eax, 23 – corect
- mov eax, [23] – funcționează
- mov edx, [ecx + 2 \* esp + 7] – greșit (esp nu poate fi index)
- mov eax, [ebx \* 3] – funcționeaza (exemplul următor)
- mov eax, [ebx + ebx \* 2] – echivalent cu anteriorul
- mov [eax \* 9 + 12] – merge
- mov eax, [ebp \* 7] – nu mai funcționează
- mov eax, [esp \* 5] – greșit (esp nu poate fi index)
- mov eax, [ebx – edx] – greșit (nu avem voie “-”)
- mov eax, [ebx + 2]
- mov eax, ebx + 2 – syntax error (nu poti să construiesti o expresie formată dintr-un registru \* ceva)