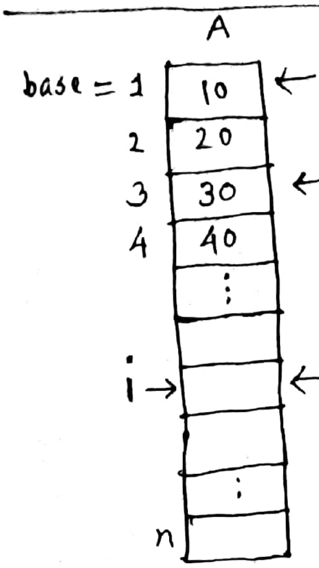
Memory Allocation

One dimensional array:



← loc(A[1]) = known

← loc(A[3]) = ?

← loc(A[i]) = ?

Each integer data = 4 bytes
= width of int
= ω

$$\text{loc}(A[3]) = \text{loc}(A[1]) + 4 + 4$$
$$= \text{loc}(A[1]) + 4*2$$
$$= \text{loc}(A[1]) + 4*(3-1)$$

In general,

$$\text{loc}(A[i]) = \text{loc}(A[1]) + \omega*(i-1)$$
$$= \text{loc}(A[\text{base}]) + \omega*(i-\text{base})$$

$$\boxed{\text{loc}(A[i]) = \text{loc}(A[\text{base}]) + \omega*(i-\text{base})}$$

Example

double A[100];
if loc(A[10]) = $(5CDF)_{16}$, find loc(A[60]).

Ans:

For double, ω = 8 bytes

$$\text{loc}(A[10]) = (5CDF)_{16}$$

i = 60

base = 10

From Formula,

$$\text{loc}(A[i]) = \text{loc}(A[\text{base}]) + \omega*(i-\text{base})$$
$$\Rightarrow \text{loc}(A[60]) = \text{loc}(A[10]) + 8*(60-10)$$
$$= (5CDF)_{16} + 8*50$$
$$= (5CDF)_{16} + (400)_{10}$$
$$= (5CDF)_{16} + (190)_{16} = (5E6F)_{16}$$

$$\begin{array}{r|l} 16 & 400 \\ \hline 16 & 25-0 \\ \hline 16 & 1-9 \\ \hline & 0-1 \end{array}$$

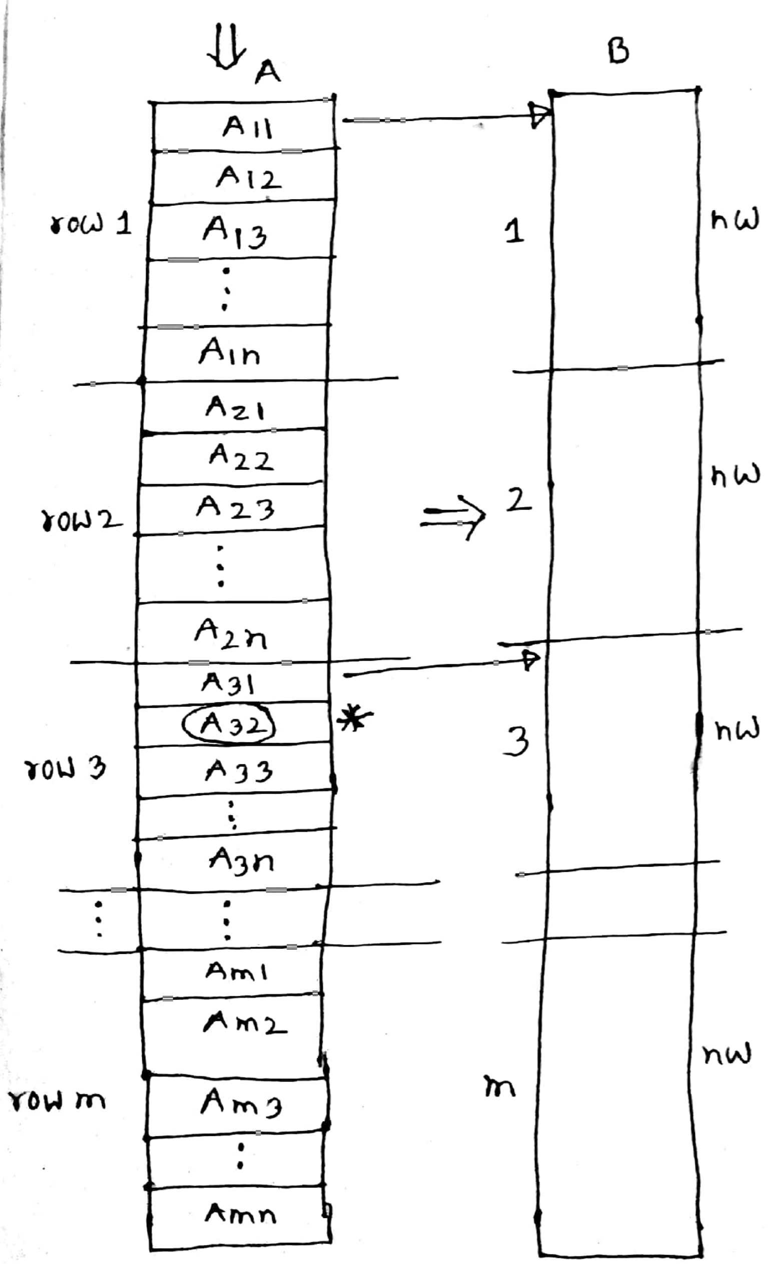
$$\begin{array}{r} 5CDF \\ 190 \\ \hline 5E6F \end{array}$$

Two dimensional Array:

Row major Allocation:

$$\begin{array}{l|cccccc|}
\text{row}1 \rightarrow & A_{11} & A_{12} & A_{13} & \cdots & A_{1n} \\
\text{row}2 \rightarrow & A_{21} & A_{22} & A_{23} & \cdots & A_{2n} \\
\text{row}3 \rightarrow & A_{31} & A_{32} & A_{33} & \cdots & A_{3n} \\
 & \vdots & & & & \\
\text{row } m \rightarrow & A_{m1} & A_{m2} & A_{m3} & \cdots & A_{mn}
\end{array}$$

One data width $= \omega$



$$loc(B[i]) = loc(B[\overset{base}{1}]) + nw(\overset{base}{i}-1)$$

$$\Rightarrow loc(B[3]) = loc(B[1]) + nw(3-1)$$

$$\Rightarrow loc(A_{31}) = loc(A_{11}) + nw(3-1)$$

Again,

$$loc(A_{32}) = loc(A_{31}) + \omega$$

$$= \underbrace{loc(A_{31})} + \omega(2-1)$$

$$= \underbrace{loc(A_{11}) + nw(3-1)} + \omega(2-1)$$

$$= loc(A_{11}) + \omega[n(3-1)+(2-1)]$$

In general,

$$loc(A_{ij}) = loc(A_{11}) + \omega[n(i-1)+(j-1)]$$

For row-wise allocation

$$\boxed{loc(A_{ij}) = loc(A_{11}) + \omega[n(i-1)+(j-1)]}$$

for column-wise allocation

$$\boxed{loc(A_{ij}) = loc(A_{11}) + \omega[m(j-1)+(i-1)]}$$

Example

float $A[\overset{m}{10}][\overset{n}{20}]$;

if $loc(A[0][0]) = (5CDF)_{16}$, Find $loc(A[\overset{i}{6}][\overset{j}{15}])$.

Assume row-major allocation

Ans:

For float, $\omega = 4$ bytes

$loc(A[0][0]) = (5CDF)_{16}$

$i = 6$

$j = 15$

$m = 10$

$n = 20$

From formula,

$$loc(A[i][j]) = loc(A[0][0]) + \omega[n(i-0) + (j-0)]$$

$$= (5CDF)_{16} + 4[20(6-0) + (15-0)]$$

$$= (5CDF)_{16} + 4[20 * 6 + 15]$$

$$= (5CDF)_{16} + 4[120 + 15]$$

$$= (5CDF)_{16} + 4 * 135$$

$$= (5CDF)_{16} + (540)_{16}$$

$$= (5CDF)_{16} + (21C)_{16}$$

$$= (5EFB)_{16}$$

$$\begin{array}{r|l} 16 & 540 \\ \hline 16 & 33 - \textcircled{12}\ C \\ \hline 16 & 2 - 1 \\ \hline & 0 - 2 \end{array}$$

$$\begin{array}{r} 5CDF \\ 21C \\ \hline 5EFB \end{array}$$

Memory Allocation for structure :

```
struct student {
  int id1;
  int id2;
  char a;
  char b;
  float percentage;
};
```

Find Memory allocatéon for this structure. Assume that 32 bit computer.

Ans:

From structure

id1	id2	a	b	percentage
← 4 byte →	← 4 byte →	1 byte →	← 1 byte →	← 4 byte →

⇒ Total 14 bytes

Since 32 bit computer

Word size = 32 bit
= 4 × 8 bit
= 4 × 1 byte = 4 byte

First cpu cycle = id1 (4 byte)

second cpu cycle = id2 (4 byte)

3rd cpu cycle = a, b, empty (4 byte)
↓ 1, ↓ 1, ↓ 2 bytes

4th cpu cycle = percentage (4 byte)
↓ 4 byte

Actual structure Memory allocation

structure padding

id1	id2	a	b	empty	percentage
4 bytes	4 bytes	1	1	2	4 bytes
		4 bytes			

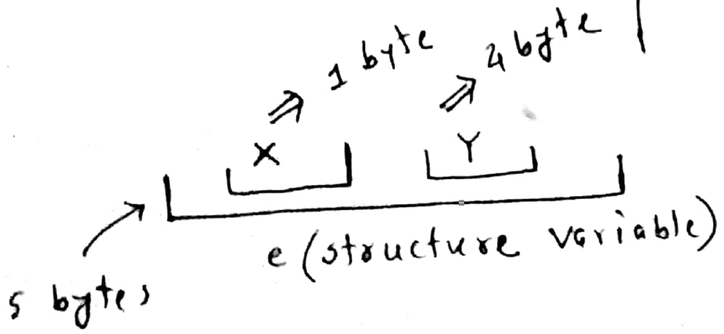
Total 16 bytes

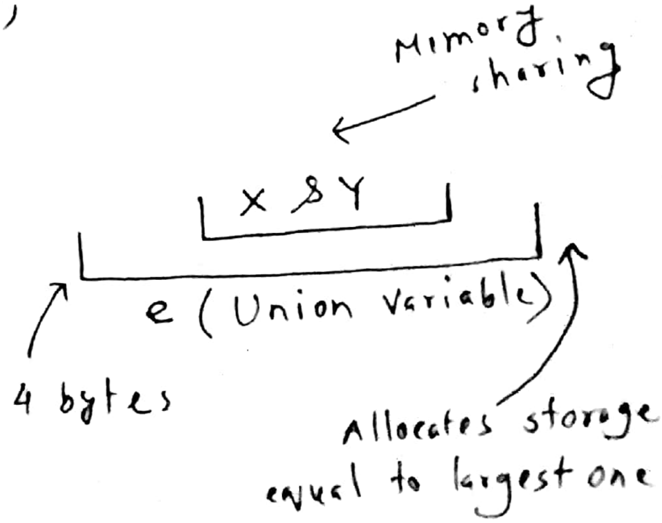
Data types	Memory allocation in C 32 bit computer			
	From Address	To Address	Total bytes	
int id1	675376768	675376771	4	→ 1 cpu cycle
int id2	675376772	675376775	4	→ 1 cpu cycle
char a	675376776	675376776	1	→ 1 cpu cycle
char b	675376777	675376777	1	
structure Padding (675376778 to 675376779) empty			2	
float percentage	675376780	675376783	4	→ 1 cpu cycle

Difference between Memory Allocation of structure and union:

```
struct Emp
{
 char x;    // 1 byte
 float Y;   // 4 byte
}e;
```

```
union Emp
{
 char x;
 float Y;
}e;
```



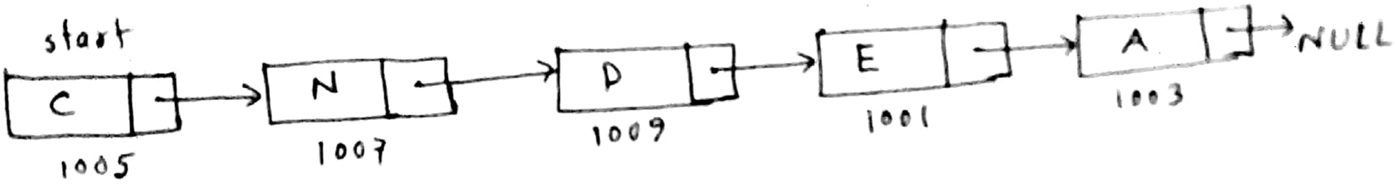


Linked list:

Address	data	next
1001	E	1003
1002		
1003	A	NULL
1004		
1005	C	1007
1006		
1007	N	1009
1008		
1009	D	1001
1010		

start

1005


start
C
1005
N
1007
D
1009
E
1001
A
1003
NULL