Agenda

1D Array
Pointer Arithmetic

Array can be implemented to ways:
1. Static implementation of array - Early Binding
   a. Memory is resolved at compile time
   b. To be used when programmer is aware of how many bytes are required in advance
   c. Memory once given cannot be shrinked or grown at runtime
   d. Memory is not in control of programmer
2. Dynamic Array - Late binding
   a. Memory will be resolved at runtime
   b. Such memory can be shrinked or grown at runtime
   c. Memory is in control of programmer
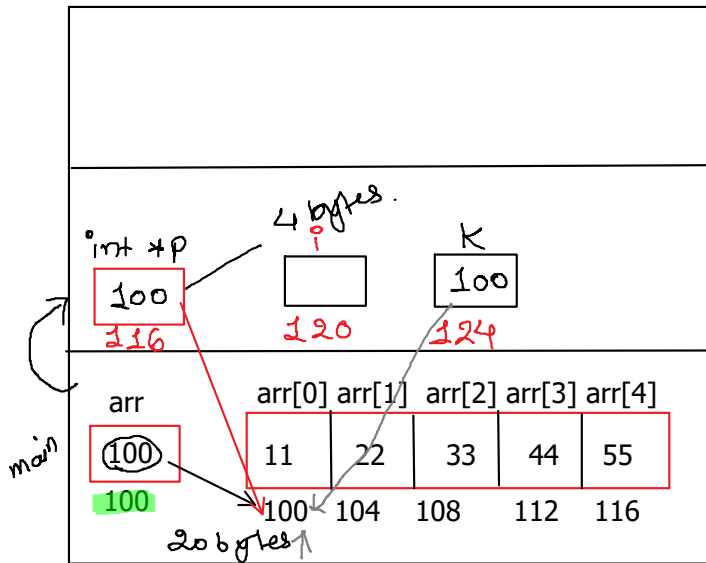
index value

ptr to int
add of int

constant

arr

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
11 int	22 int	33 int	44 int	55 int

100

100

100

104   108   112   116

addef array

Sizeof (arr)

accept

main
array

display
resel.
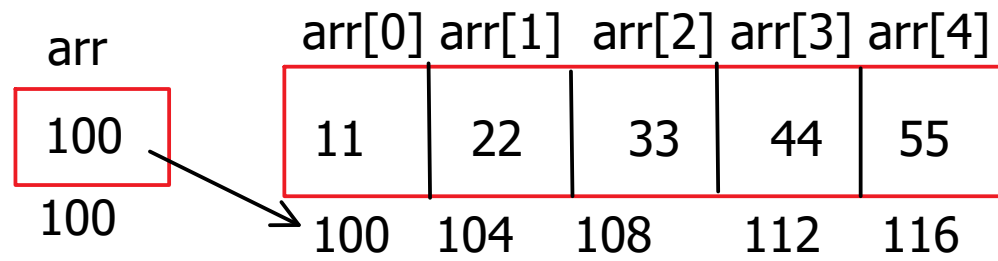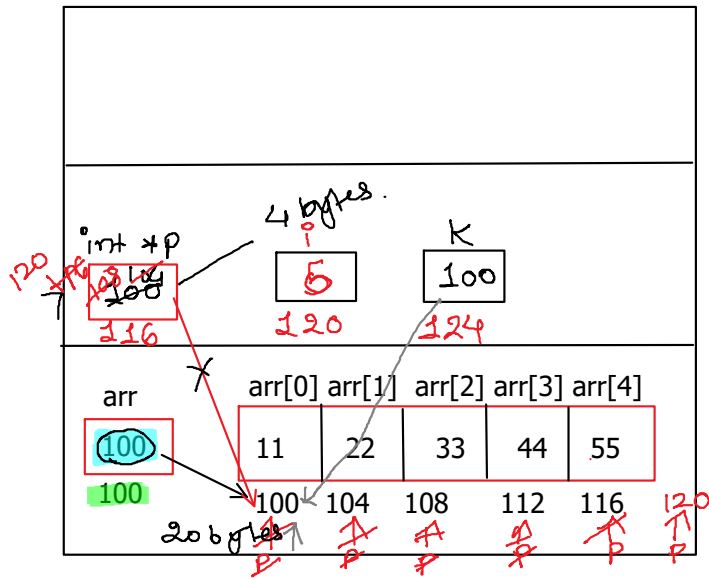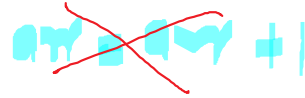
arr + 1   100 + (*4)   = 104
                 add of-   int
                  int   const
                 Scale factor
                 of int
int *k = P;   4 bytes.

&arr + 1

add of
Array
↓
Scale factor
20
         pointer

100 + 1
add of int   elel

int *P   4 bytes.   K
100   100
116   120   124

arr   arr[0] arr[1] arr[2] arr[3] arr[4]
main   100   11 | 22 | 33 | 44 | 55
       100   100   104   108   112   116
       20 bytes

*(arr + 3)

arraynotation
        arr[3]   3[arr]   *(arr + 3)

*(arr + 3)   *(3 + arr)

arr

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
11	22	33	44	55

100

100

100 104 108 112 116

$P++ \quad = \quad P = P+1$

$P = 100 + 1$

104

$P++ = P = P+1$

$P = 104 + 1 = 108$

$\vdots$

$P++ = 112 + 1 = 116$

$P[-1] \qquad *(P-1) \qquad *(120-1)$

addr.

int

$*(116) = 55$

Diagram area (left):

int *P

4 bytes.

$\boxed{5}$ K $\boxed{100}$

120   124

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
11	22	33	44	55
100	104	108	112	116

arr

100

100

20 bytes