

Agenda

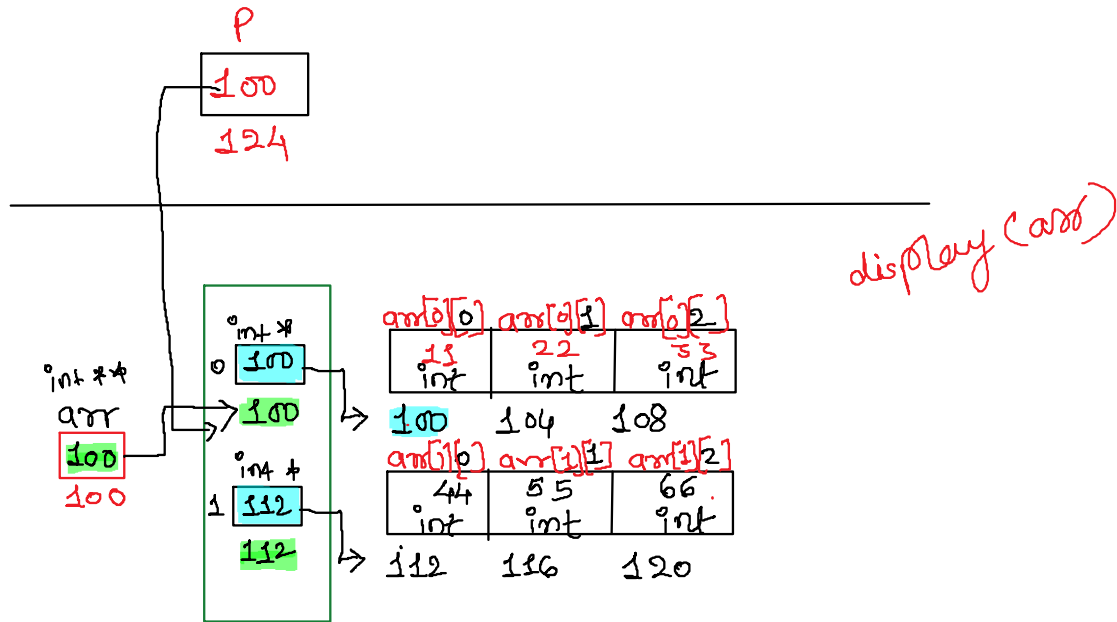
Monday, May 22, 2023 8:15 AM

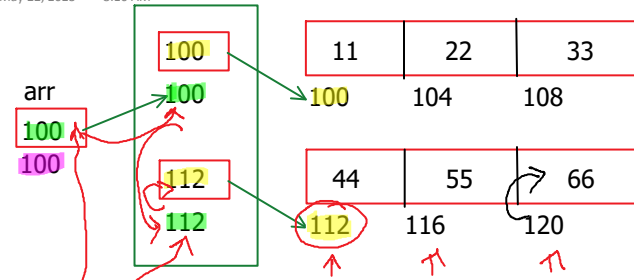
2D Array

Pointer Arithmetic with 2D Array

Dynamic Memory Allocation

`int arr[2][3]`
row
Student-
cols
marks





arr --> 100 --> is address of row / address of pointer to int
 arr+1 = 100 + 1 --> 112 //address of row/address of pointer to int
 *(arr+1) == 100+1 == 112 ==> *112 ==> 112 ==> address of int/address of col
 *(arr+1)+2 == 100+1 == 112 ==> *112 ==> 112 ==> 112+2 == address of int /address of col
 (*(arr+1)+2) == 100+1 == 112 ==> *112 ==> 112 ==> 112+2 = 120 ==> *120 = 66 //int value
 &arr+1 == 100 ==> address of array (2D) ==> 100+1 == 124

$$\begin{aligned} &arr + 1 \\ &100 + 1 \\ &\underline{\quad} \\ &112 \end{aligned}$$

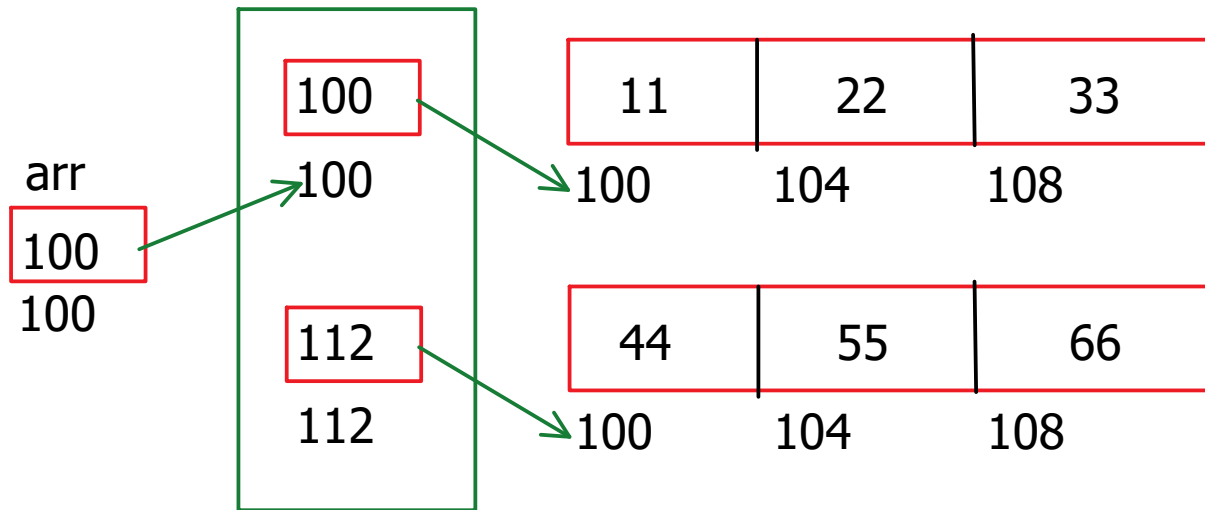
$$\begin{aligned} &arr + 1 \\ &(100 + 1) \\ &\underline{*} \\ &112 \\ &\underline{\quad} \\ &112 \end{aligned} + 2$$

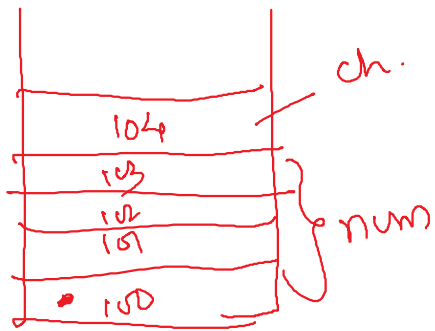
$$120$$

$$*(arr+1) = \text{addr of int}$$

$$(*(arr+1)) + 2$$

$$*(*(arr+1) + 2) = \text{int value}$$



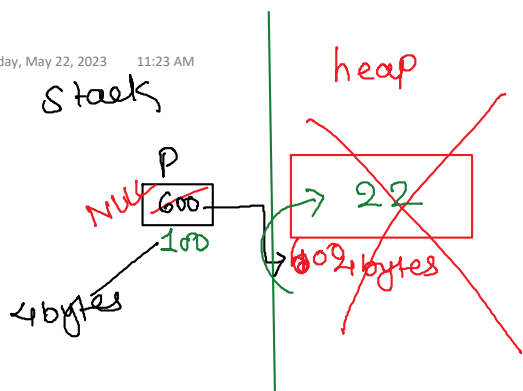


$c = \&ch$. *c \rightarrow 1 bytes.

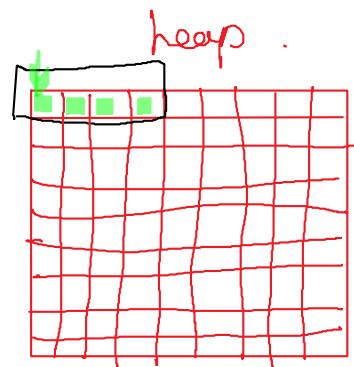
00000000 00000000 00000011 00010101
 103 102 101 100
 ↑

$\&num = 100$

$p = 100$ *p
 ↓
 4 bytes \leftarrow



free(P)
free(600);



600...603 4 bytes.
p = (int *) malloc(sizeof(int));

malloc(sizeof(int)) malloc(4)
calloc(1, sizeof(int)) calloc(1, 4)
no. of eles size of each ele

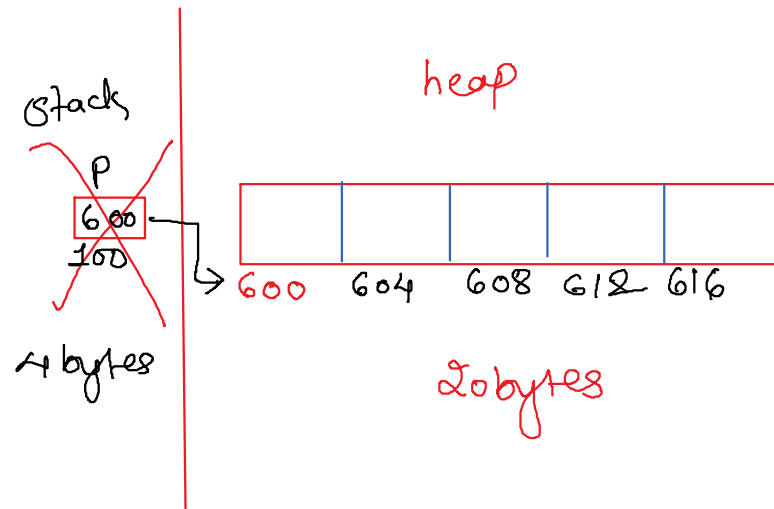
malloc(sizeof(int) * 5);

calloc(5, sizeof(int)); calloc(5, 4);

```
void fun()
{
    int *p;
    p = (int *) malloc(sizeof(int)*5);
}
```

Handwritten annotations:

- A red box around the `malloc` call with `600` written above it.
- A red wavy line under the `malloc` function name.
- `malloc(20)` written below the `malloc` call.



Handwritten: `int arr[5]`

