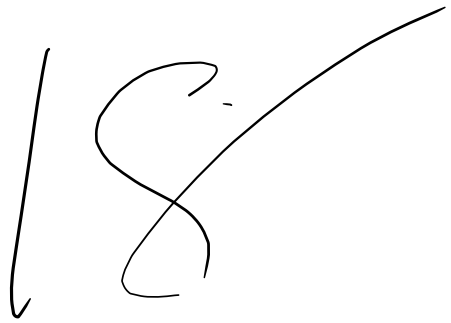
Let’s solve it



Array

List of elements

int arr1[10]; number

How many?

Fix size of array

are called static arrays;

(compile time)

Variable Length Array (VLA)

```
int n;
scanf("%d", &n);
int arr[n];
```

(Advanced Topic)

Dynamic array (will learn later)

Types of ways

Various ways to answer

1st Way → Compile time / Runtime
static / Dynamic

2nd way

✓ int arr[10]; Integer A

✓ char arr2[20]; char A

(Ragged arrays?)

3rd way — will it have ~~similar~~/same data type of all elements
→ Homogenous ✓ ——

4th way — sparse array Vs dense array
(~~forest~~)

5th way dimensions
- 1-D array, 2-D array, 3-dimensional

Arrays are nothing but special variables. And hence, I can have multiple arrays in my program.

Array of 10 numbers

input

```
int main()
{  int input; int i; int sum=0;
   for(i=1; i<=10; i++)
   {  scanf("%d",&input);
   }  sum += input;
   printf("%f", sum/(float)10);
```

Do I have all input available?

To have 10 input locations

int n1, n2, n3, n4, n5, n6, n7, n8, n9, n10;

scanf("%d %d %d ... %d", &n1, &n2, ..., &n10);

10 times

Above is tedious.

int input[10];

Size / capacity

Array

Index of element

scanf("%d", &input[0]);

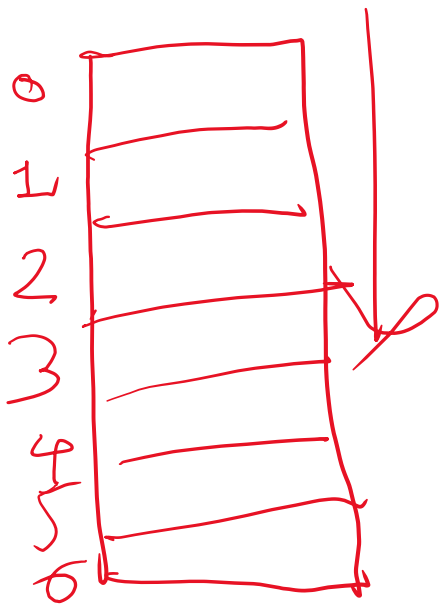
Index starts from 0.

[1]
[2]
:
[9]

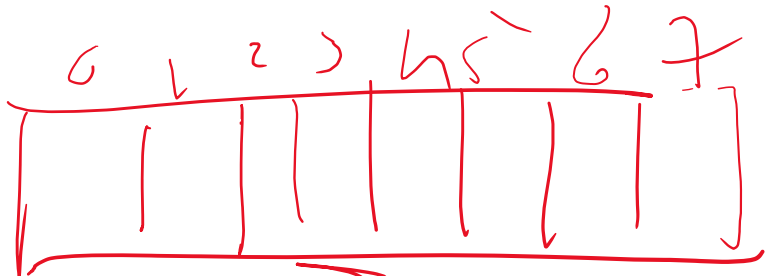
The first element it points an [0]

arr[3] read as arr of 3
for actually 4th element.

int data[7];



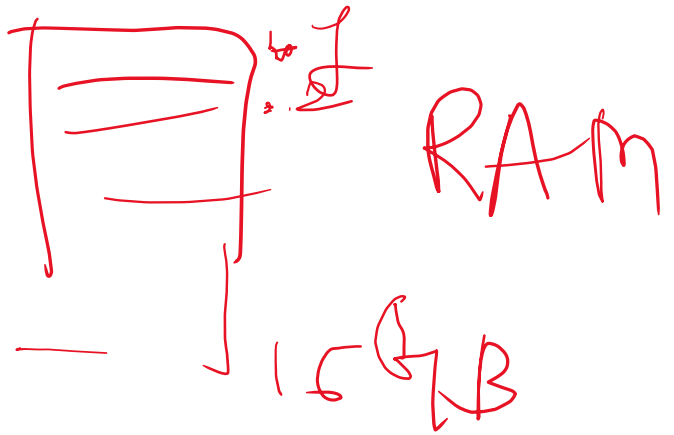
Vertically



int data[8];

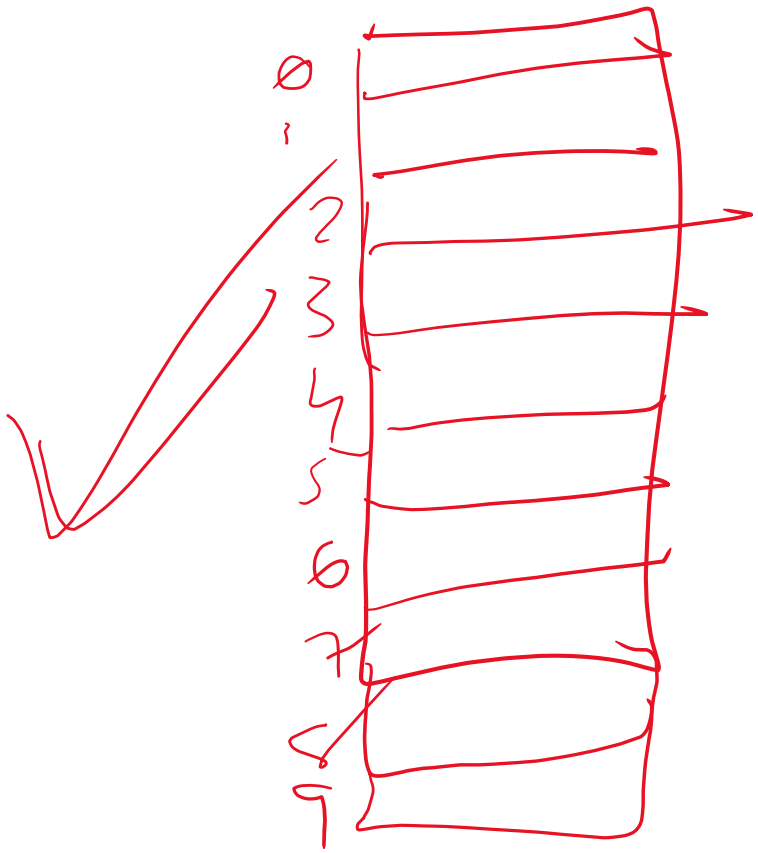
Array name is a constant pointer to 0th element. Base address.

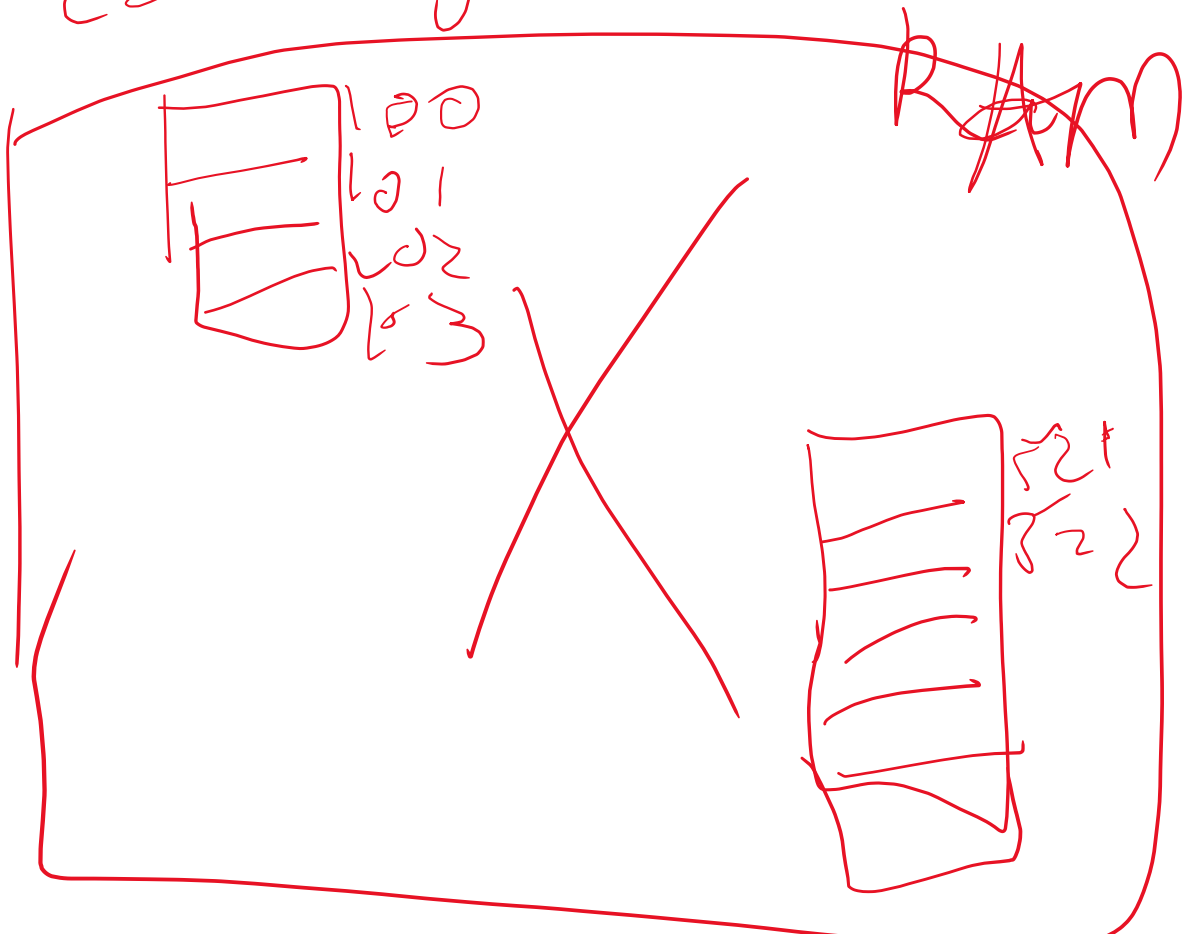
In computer memory, all bytes are referred using a number called "address". Virtual



How array works? given

All elements of an array will always be in contiguous memory.





There is maths for referring to elements via index (imagine 2 bytes

int arr[5]; → size or capacity

element

locate element (is of two bytes)

arr[0] to 4 (5-1)

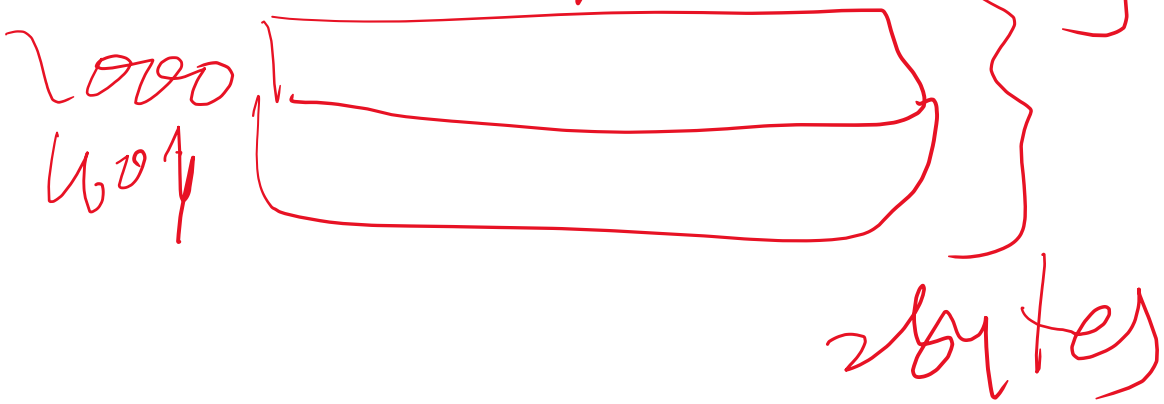
base address + (0 * sizeof(int))

arr[0] ⇒ arr + (0 * sizeof(int))

⇒ arr + 0

⇒ arr

{ Read int from address starting from 1000 }



arr[i] → arr + (i * sizeof(int))

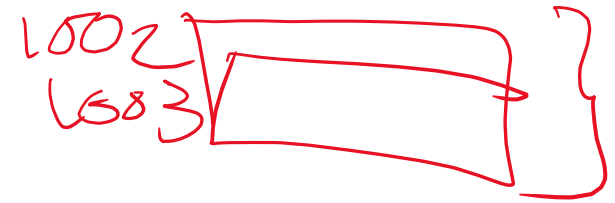
+ 2

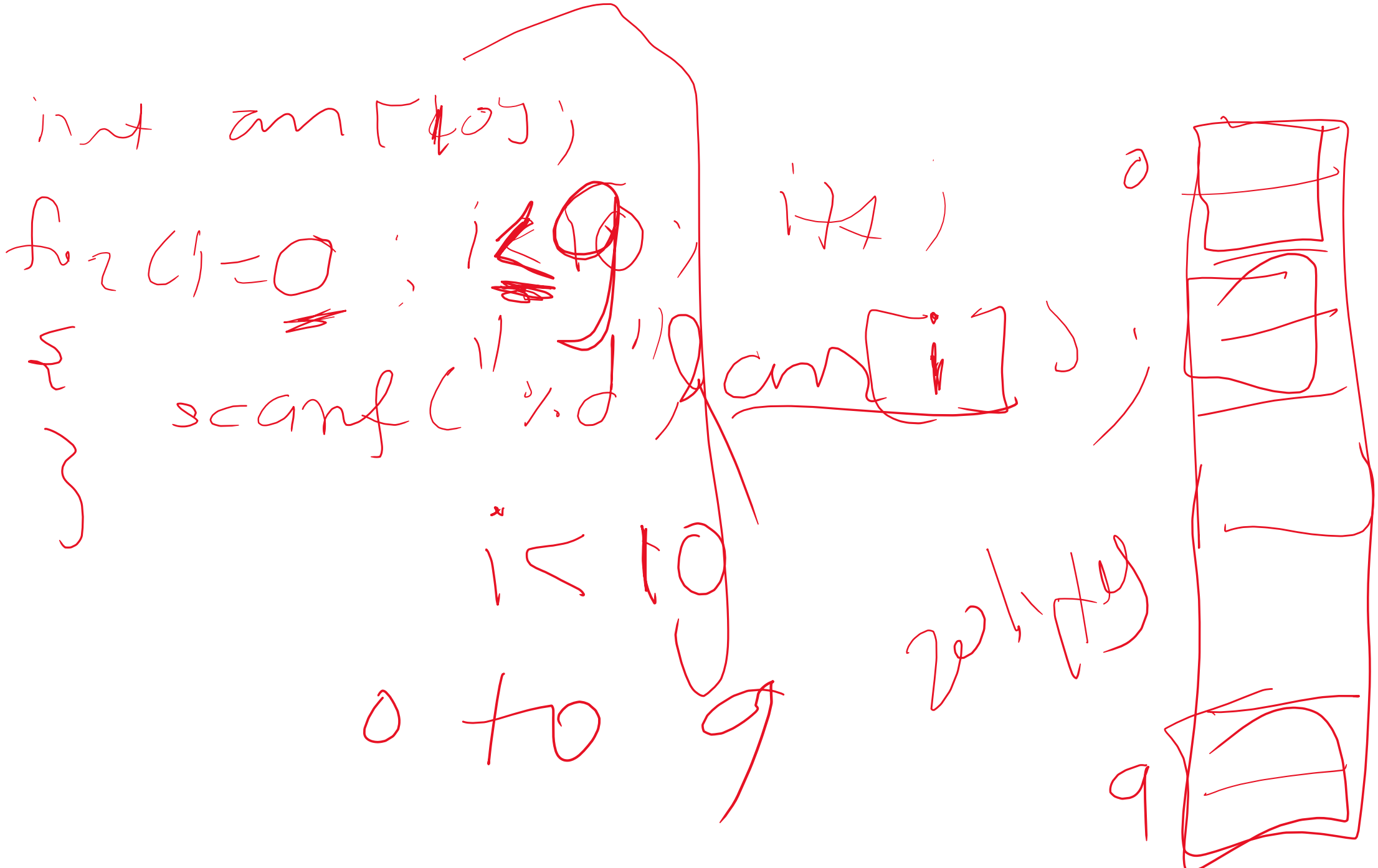
1000 + 2

1002

Read element starting

int from address 1002




int arr[10];
for(i=0; i<10; i++)
{
scanf("%d",&arr[i]);
}
i<10
0 to 9
0
9