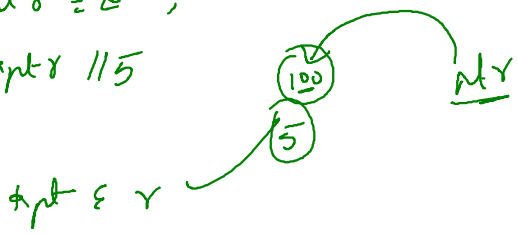
Pointers with Arrays

```
int x = 5;
cout<<x;    //5
  &x;       //100
int* ptr = &x;
cout<<*ptr  //5.
cout<<ptr   //100
```

Lecture 18 //

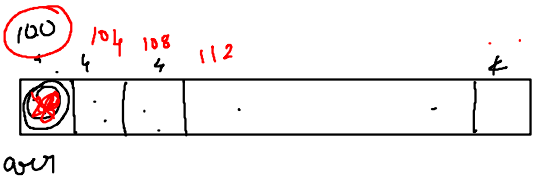
```
*ptr = &x;
cout*ptr //5
```



Declaring array 4bytes

```
int arr[10];
```

4 x 10 => 40 bytes



arr → Address of first block.

arr[0] //value

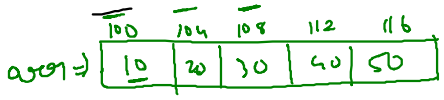
&arr[0] //

arr == &arr[0] //

```
int x=10
int* ptr = &x;
cout<<*ptr
```


```
cout<<arr
cout<<*arr.  // 10 //
```

first MB



cout<<arr ⟶ 100.

cout << &arr // 10

cout << &(arr+1) (100+1) = 101. error

(100 + 4) .

* 104. // 20 //

(arr+3) => 108

30 //

*arr + 1

10 + 1 => 11

Symbol table

arr → 100 → fixed

cout << arr+1 — 100+4 => 104

arr = arr + 1 Error.

=> 100 + 4

arr → 104

int *ptr = arr;

cout << *ptr.; //10

*ptr = *ptr + 1

*ptr => 10 + 1 //

=> arr[0] //

?

10+1 (11) //

100

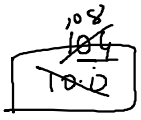
X arr = arr+1 → ST. Error.

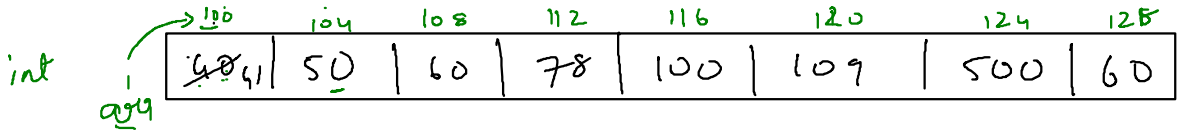
✓ ptr = ptr + 2 → Yes.

*ptr // 20

// 30

or





cout << arr // 100.

cout << &arr[0] // 100

cout << arr[0] // 40

cout << *arr // 40

cout << *arr + 1 // 41

cout << *(arr + 1) // 50

int *ptr = arr;

cout << *ptr // 40

*ptr = *ptr + 1

=> 41

ptr = ptr + 1

100 + 4

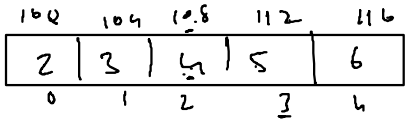
ptr -> 104

cout << *ptr // 50 //.

ST

arr -> 100

Formula -



arr[2] => *(arr + 2) // 4 = 4

arr[3] => *(arr + 3) // 5 = 5

100+8 =108

100+12
112

arr[i] = *(arr + i) -> formula

i[arr] = *(i + arr)

Different pointers & arr

int arr[10]; 10×4 => 40 bytes

4 bytes.

int *ptr = arr; -> 8 bytes.

4 bytes

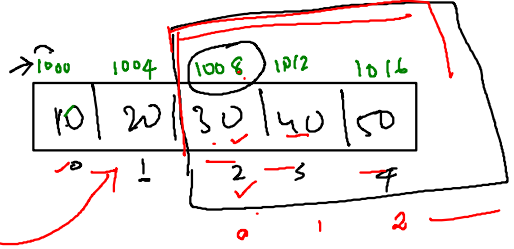
8 bytes.

— X —

-> Pointer in function call.

Lecture 19.

```
print(int *arr) {
    cout << arr[0] }        // 10  1130
main() {        arr[1] }    // 20  1140
int arr[5] = {10,20,30,40,50};
print(arr)
print(arr+2)
```



arr[:]

o/p 20 H/w.

1000 + 8
1008

+arr ≡ arr[]

arr[] ⇒ +arr

(arr, n)

① Sum of all digits.

arr[5] = {7, 8, 1, 2, 3}

7+8+1+2+3 ⇒ 21 //

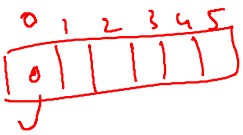
```
main() {
int arr[5] = {7,8,1,2,3};          // 20 //4
int size = sizeof(arr) / sizeof(arr[0]);
printSum(arr, size);
}
```

→arr

```
void printSum(int arr[], int size) {
    int sum = 0;
    for(int i=0; i<size; i++) {
        sum = sum + arr[i];
    }
    cout << sum;
}
```

5

0 1 2 3 4

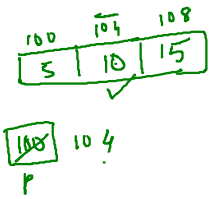


7+8

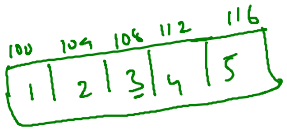
i=0
sum = 7
i=1
sum = 15
i=2
sum = 16
i=3
sum = 18
i=4
sum = 21
i=5

Practice Problems

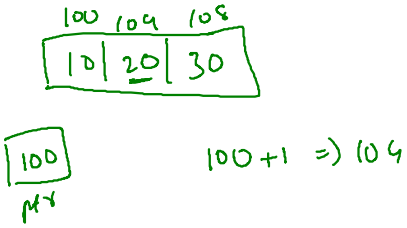
Q1: What is the output?
```
int arr[3] = {5, 10, 15};
int *p = arr;
p++;
cout << *p;
```
a) 5
b) 10
c) 15
d) Garbage



Q2: What is the output?
```
int arr[5] = {1, 2, 3, 4, 5};
cout << *(arr + 2);
```
a) 2
b) 3
c) 4
d) 5

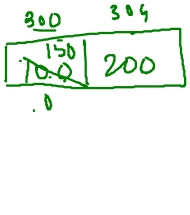


Q3: What is the output?
```
int arr[] = {10, 20, 30};
int *ptr = arr;
cout << *(ptr + 1);
```
a) 10
b) 20
c) 30
d) Compilation error

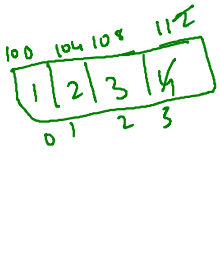


Q4: What is the output?
```
int arr[4] = {7, 14, 21, 28};
cout << arr[2] + *(arr + 1);
```
a) 35
b) 28
c) 21
d) 14

Q5: What is the output?
```
int arr[2] = {100, 200};
int *p = arr;
*p = *p + 50;
cout << arr[0];
```
a) 100
b) 150
c) 200
d) 250
Output = 150

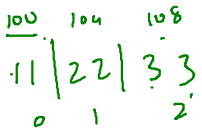


Q6: What is the output?
```
int arr[] = {1, 2, 3, 4};
int *p = arr + 1;
cout << *(p + 2);
```
a) 1
b) 2
c) 3
d) 4



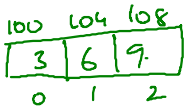
Q7: What is the output?
```
int arr[] = {10, 20};
int *p = arr;
cout << *(p++) + *p;
```
a) 20
b) 30
c) 40
d) Undefined

Q8: What is the output?
```
int arr[3] = {11, 22, 33};
cout << *arr << " " << *(arr + 2);
```
a) 11 22
b) 11 33
c) 22 33
d) 33 11



11 33

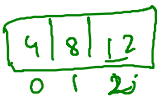
Q9: What is the output?
```
int arr[3] = {3, 6, 9};
int *ptr = arr;
ptr += 2;
cout << *ptr;
```
a) 3
b) 6
c) 9
d) Compilation Error



arr[i] = *(arr + i)

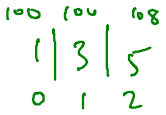
Q10: What is the output?
```
int arr[] = {4, 8, 12};
cout << 2[arr];
```
a) 4
b) 8
c) 12
d) Compilation Error

2[arr]

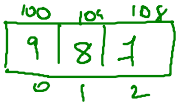


x i[arr] = *(i + arr)

Q11: What is the output?
```
int arr[] = {1, 3, 5};
int *ptr = arr;
cout << *(++ptr);
```
a) 1
b) 3
c) 5
d) Error

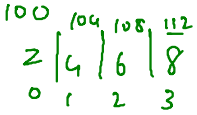


Q12: What is the output?
```
int arr[] = {9, 8, 7};
int *p = arr + 1;
cout << *(p - 1);
```
a) 9
b) 8
c) 7
d) Garbage

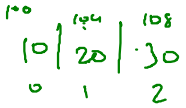


Q13: What is the output?
```
int arr[] = {2, 4, 6, 8};
cout << *(arr + 4);
```
a) 2
b) 4
c) 8
d) Garbage value

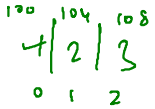
100 + 16 =) 116



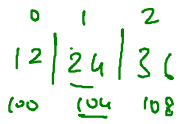
Q14: What is the output?
```
int arr[] = {10, 20, 30};
int *ptr = arr;
ptr = ptr + 1;
cout << *ptr;
```
a) 10
b) 20
c) 30
d) Compilation Error



Q15: What is the output?
```
int arr[] = {1, 2, 3};
int *ptr = arr;
cout << *(ptr + 0);
```
a) 0
b) ~~1~~
c) 2
d) 3

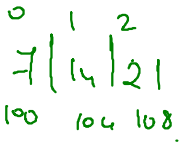


Q16: What is the output?
```
int arr[] = {12, 24, 36};
cout << arr[1] + *(arr + 1);
```
a) 24
b) 36
c) ~~48~~
d) 60

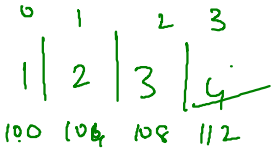


24 + 24

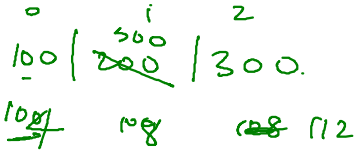
Q17: What is the output?
```
int arr[3] = {7, 14, 21};
int *ptr = &arr[0];
cout << *(ptr + 2);
```
a) 7
b) 14
c) ~~21~~
d) Garbage



Q18: What is the output?
```
int arr[4] = {1, 2, 3, 4};
int *ptr = arr + 3;
cout << *ptr;
```
a) 1
b) 2
c) 3
d) ~~4~~

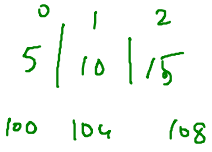


Q19: What is the output?
```
int arr[3] = {100, 200, 300};
int *ptr = arr;
ptr++;
*ptr = 500;
cout << arr[1];
```
a) 100
b) 200
c) ~~500~~
d) 300



Q20: What is the output?
```
int arr[] = {5, 10, 15};
int *p = arr;
cout << *p + *(p + 2);
```
a) 15
b) ~~20~~
c) 25
d) 30

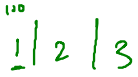
5 + 15



Example 1:

```
void print(int *arr) {
  cout << arr[0];
}
int main() {
  int a[] = {1, 2, 3};
  print(a);
}
```

1



Example 2:

```
void printSecond(int *arr) {
  cout << arr[1];
}
int main() {
  int a[] = {10, 20, 30};
  printSecond(a);
}
```

// 20

Example 3:

```
void sum(int *arr, int n) {
  int total = 0;
  for(int i = 0; i < n; i++) total += arr[i];
  cout << total;
}
int main() {
  int a[] = {2, 4, 6};
  sum(a, 3);
}
```

// 12

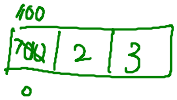
3

total = 0
i = 0
total = 2
i = 1
total = 6
i = 2, 2 < 3
total = 12

i = 3 3 < 3 ✗

Example 4:

```
void modify(int *arr) {
  arr[0] = 100;
}
int main() {
  int a[] = {1, 2, 3};
  modify(a);
  cout << a[0];
}
```



// 100

```
void partialPrint(int *arr) {
  cout << arr[1] << " " << arr[2];
}
int main() {
  int a[] = {5, 10, 15};
  partialPrint(a)
}
```

// 10 15