

Chapter - 8 :- Pointers :-

Normal variable `int a;`
 `a = 5;`

a 5
1000

Defining pointer

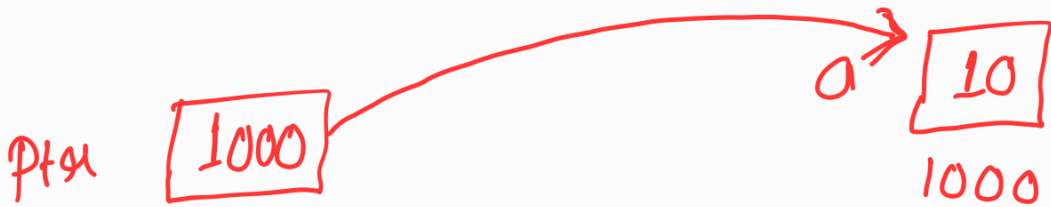
data-type *variable-name;

Ex:- `int *a;`
 ↑ pointer variable.

Initializing pointer :-

Initializing pointer :-
 int a, *ptr; ptr → pointer variable.

$P_{th} = 2a; \quad a = 10;$



* push ← dereferencing.

 $\times(2a)$

a

$$p + H = 2a$$
$$* p + q = 50$$

✓
x(20)
a

a $\boxed{50}$
1000

```
int a = 10, *ptr;
```

```
ptr = &a;
```

a 10 1000
ptr ↑

```
printf("value of a = %d \t address of a = %u \n",  
a, &a);
```

```
printf("value of a = %d \t address of a = %u \n",  
*ptr, ptr);
```

(#) Pointed in expression.

```
C = a + b;
```

```
int a, b, c, *ptr1, *ptr2;
```

```
a = 10;
```

```
b = 5;
```

```
ptr1 = &a;
```

```
ptr2 = &b;
```

```
C = *ptr1 + *ptr2;
```

*ptr1
a

*ptr2
b

// Array & pointer

```
int a[5];
```

1000

1004

1008

1012

1016

a

3	8	5	6	10
---	---	---	---	----

⇓

a[0]

a[1]

a[2]

a[3]

a[4]

= base address

= address of a[0]

= 1000



input(a, n)
↑ ↑
array int
 number

int *ptg;
ptg = a; ✓
 ↘
 ptg

1000

*ptg = 20; [20, ...]

ptg = ptg + 1;

↘

ptg 1004