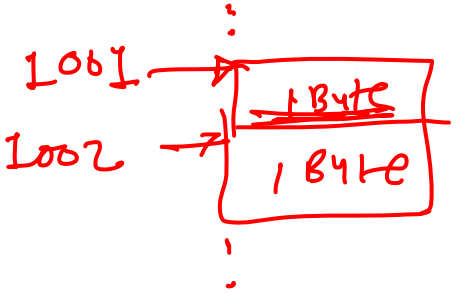
Let’s solve it

Pointers

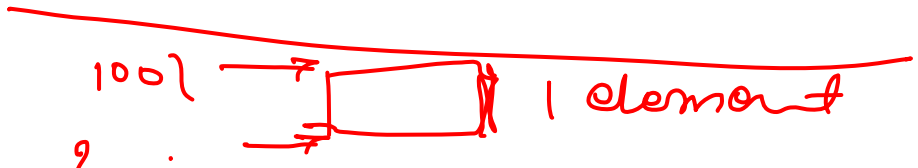
- A special type of variable bcz address of another variable.

it will store

- name, memory, value, datatype



virtual number allocated to each byte of memory is called "address"



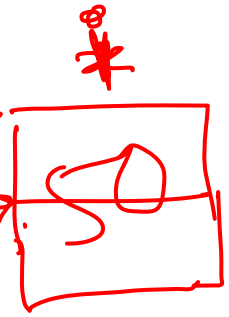
next element ?

char 1 byte
int 2 bytes
float 4 bytes

1002 (1001+1)
1003 (1001+2)
1005 (1001+4)

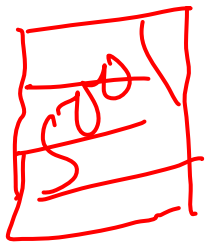
Can I create multiple pointers in my program?

```
int i = 50;
char c = 'a';
float f = 3.14159;
```



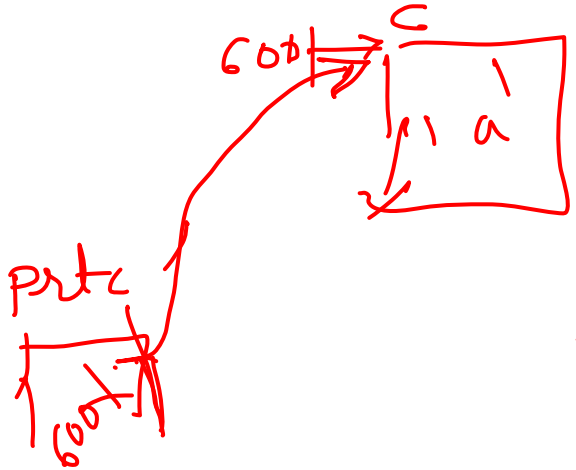
integer

```
int *ptri = &i;
```



4 bytes assumption

```
char *ptrc = &c;
float *ptrf = &f;
```



printf("%d", i);

printf("%d", *ptri);

Goto memory pointed by ptri and read data

What type of data

*ptri ⇒ type of data to be read is integer because datatype of ptri is integer

printf("%f", f);

printf("%f", *ptrf);

*ptrf ⇒ type of data to be fread datatype of ptrf is float.

printf("%c", c);

printf("%c", *ptrc);

Pointer arithmetic

Character pointer advances in the case of increment operation by sizeof(char) 5001 + 1

ptr → Integer pointer 5001 + 2 bytes

5001 | float pointer 5001 + 4 bytes

Works on element (datatype) not on a single byte itself.

--ptr; ptr--; ✓ yes

++ptr; ptr = ptr+1; ✓ yes

ptr = ptr - 1; ✓ yes

ptr = ptr / 1; ✗ Not supported.

ptr = ptr * 1; ✗ Not supported

ptr = ptr - 1;

ptr = ptr + 1;

pointer with integer constant

p3 = p1 - p2; ✓ Subtraction of two pointers are used to find no. of elements between two locations of a single array.

p1 + p2 ✗

p1 * p2 ✗

p1 / p2 ✗

two pointers

Traversing array

```
int arr[5] = {10, 20, 30, 40, 50};

int *ptr;
ptr = &arr[0]; // ptr = arr; // (array is constant pointer)
```

ptr → 1000

arr 1000

1001	10
1002	20
1003	
1004	30
1005	
1006	40
1007	
1008	50
1009	
1010	

array int 2 bytes

```
for(i=0; i<5; i++)
    printf("%d \t", arr[i]);

for(i=0; i<5; i++)
{
    printf("%d\t", *(ptr+i));
}
```

```
x = y;
ptr1 = ptr2;
```

datatype of pointer

1000 + 1 * sizeof(int)

1 * 2

{

1002

1004

1006

:

}

ptr2 = ptr2 + 1;

```
int *tptr=ptr;
for(i=0; i<5; i++)
    printf("%d\t", *(ptr++));
```

→

```
for(i=0; i<5; i++)
    printf("%d\t", *(arr+i));
```

Same

```
arr[i]
```

array with pointer notation

```
for(i=0; i<5; i++)
{
    printf("%d\t", ptr[i]);
}
```

// pointer with array[] notation

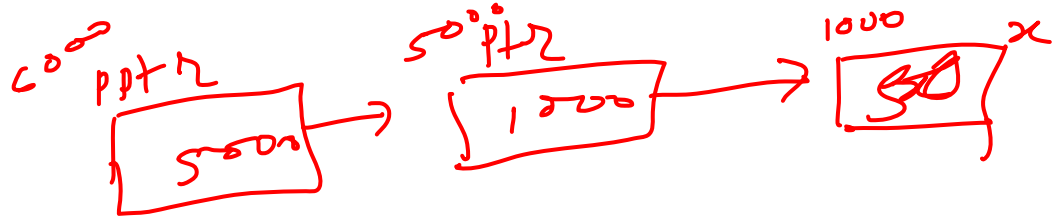
```
int *tptr = ptr;
int *lptr = ptr+5;

while ( tptr < lptr )
{   printf("%d\t", *tptr);

    tptr++;
}
```

Comparison of pointers

```
int x;
x=50;
int *ptr = &x;
int **pptr = &ptr;
printf("%d", x);
printf("%d", *ptr);
printf("%d", **pptr);
```



50

50

50

value at 1000

value at 5000

1000