```c
#include <stdio.h>
int main()
{
    char str[10];
    char *p;
    p=str;
    printf("Enter a string:");
    gets(p);
    while(*p!='\0')
    {
        printf("\n%c",*p);
        p++;
    }
    return 0;
}
```

```c
for( ; *p!='\0'; p++)
    printf("\n%c",*p);
```

Some New Shortcuts

① `if(a!=0)` ↑ optional

Same as

`or`

`if(a)`

② `while(a+b)` ⟹ `while(a+b!=0)`

{ = = = = } Same

Some More Shortcuts

① if (a == 0) or if (! a)
{
=
=
=
}
S
a
m
e

---

② if (strcmp (a, b) == 0) or if (! strcmp (a, b))
{
=
=
=
}
S
a
m
e
{
=
=
=
}

Previous Code Using Shortcut

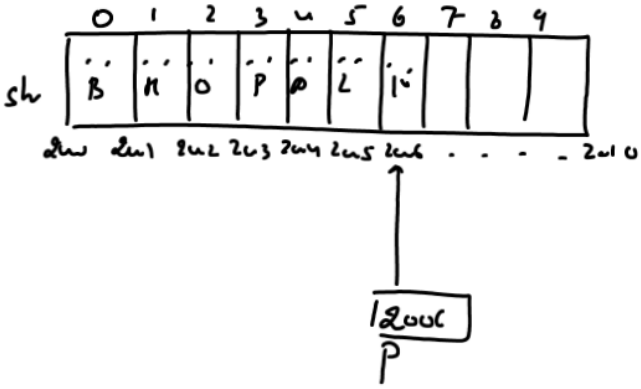
```
#include <stdio.h>
int main()
{
    char str[10];
    char *p;
    p=str;
    printf("Enter a string:");
    gets(p);
    while(*p)
    {
        printf("\n%c",*p);
        p++;
    }
    return 0;
}
```

for (; *p ; p++)
printf("\n%c", *p);

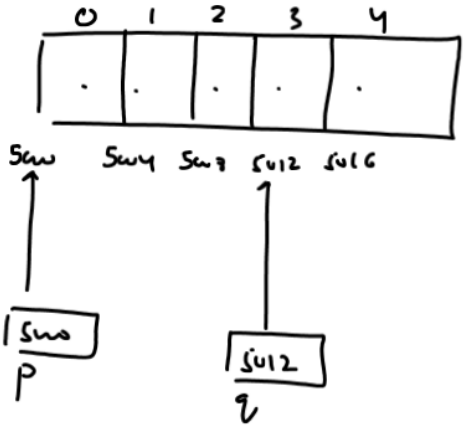
WAP to accept a string from the user and calc and print its length. You are allowed to use only:
1. A char array
2. A char *

No extra variable or any string function must be used

```
#include <stdio.h>
int main()
{
   char str[10];
   char *p;
   p=str;
   printf("Enter a string:");
   gets(p);
   while(*p)
  {
    p++;
  }
 printf("Length=%d",p-str);
 return 0;
}
```




```
int arr[5];
int *p, *q;
p = arr;
q = &arr[3];
printf("%d", q-p);
```



3

Whenever we subtract 2 addresses, the compiler always divides the result of subtraction with the size of pointers data type and then returns the result

if p and q are 2 pointers, then:

q + p X

q * p X

q / p X

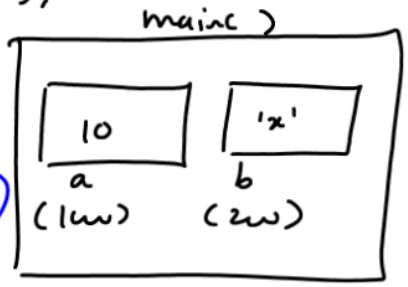
q % p X

q - p ✓] only when p & q are
p - q ✓] q same data type.

Other than subtraction, no other arithmetic operation is allowed between 2 pointers

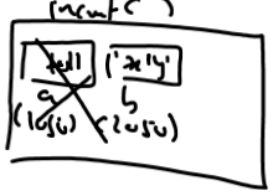
```
void increment ( int, char);

int main ( )
{
    int a = 10;
    char b = 'x';

    printf (" Before calling increment, a = %d, b = %c ", a, b);
    increment (a, b);
    printf ("\n After calling increment, a = %d, b = %c ", a, b);
    return 0;
}
void increment ( int a, char b)
{
   a++;
   b++;
}
```

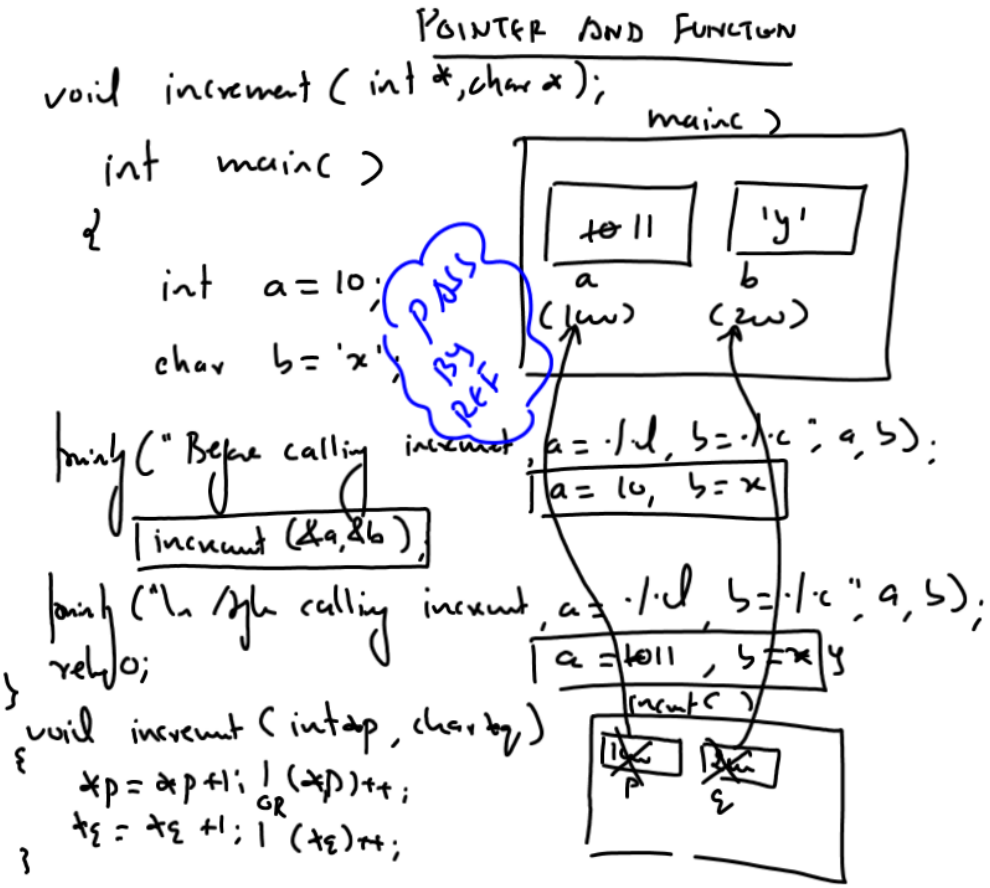
Pass by value



a = 10, b = x

a = 10 , b = x





WAP to accept 2 integers from the user and pass them to a function called swap. Inside the function exchange the values of the variables . But print back the swapped values in main()

```
void swap(int *,int *);
int main()
{
  int a,b;
  printf("Enter 2 int:");
  scanf("%d %d",&a,&b);
  printf("Before swapping a=%d,b=%d",a,b);
  swap(&a,&b);
  printf("\nAfter swapping a=%d,b=%d",a,b);
  return 0;
}
void swap(int *p,int *q)
{
  int temp;
  temp=*p;
  *p=*q;
  *q=temp;
}
```