

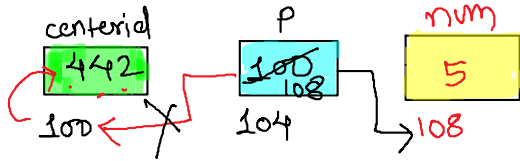
## Agenda

Qualifier - const

Basics of String

String library functions

`int *p = &centerid`

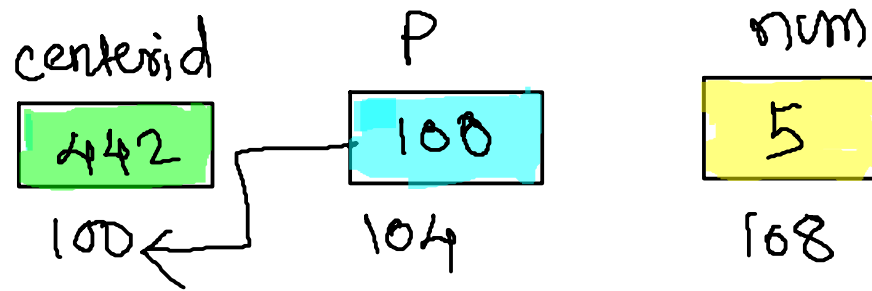


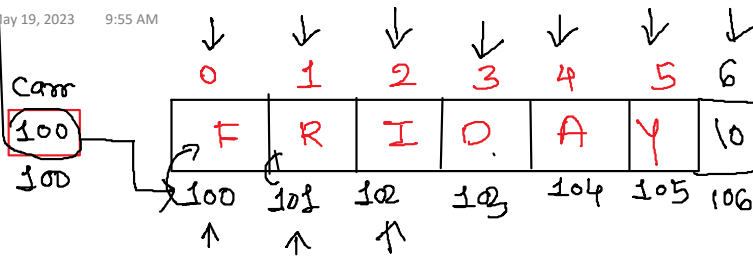
`const int *p = &centerid;`

`p = &num;`

\*p is constant

if p is pointing to centerid  
 then green block is constant via p  
 if p is pointing num now value at p  
 nothing but yellow block is const



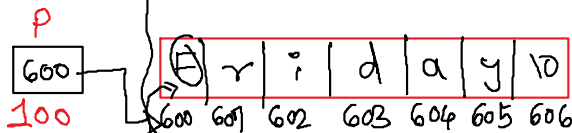


```

int i=0;
while(carr[i] != NULL)
{
    printf("%c", carr[i]);
    i++;
}
    
```

FRIDAY

char \*p = "Friday" — Handcoded Value → String literal constant



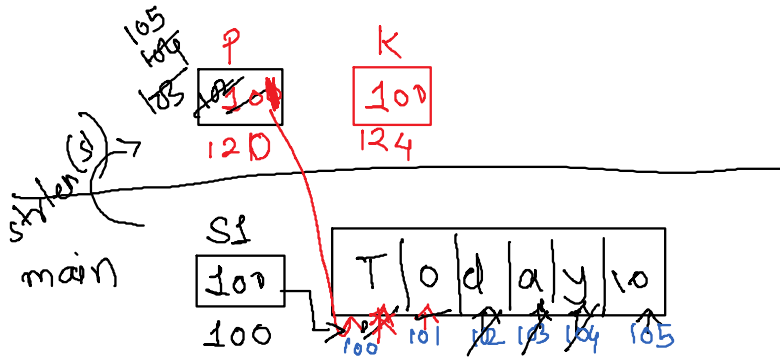
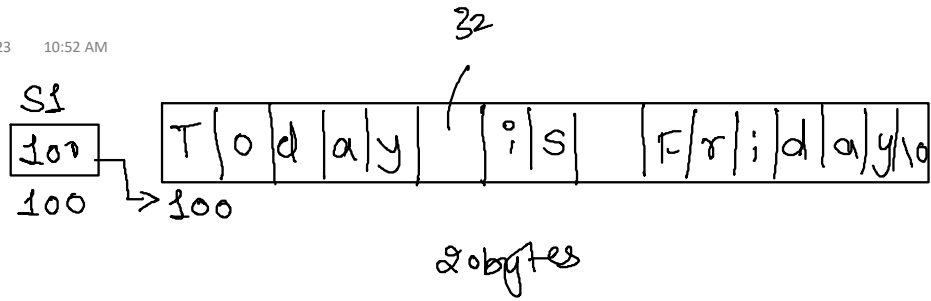
4 bytes  
stacks.

7 bytes  
Data Section - R.O - Read Only.

```

int main()
{
    char *p = "Friday";
    printf("size= %u \n", sizeof(p));
    printf("p=%u &p=%u p=%s\n", p, &p, p);
}
    
```

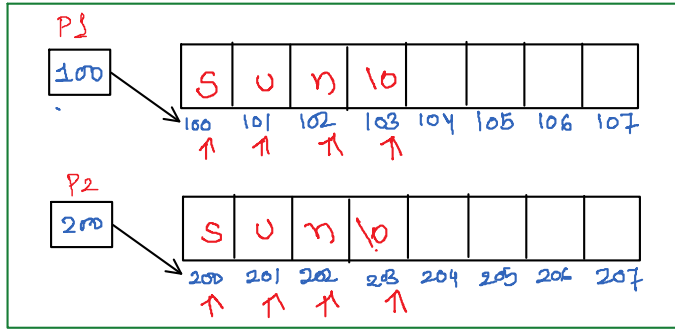
\*p = 'f';



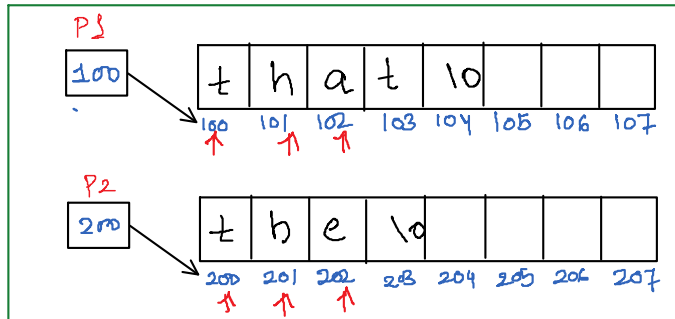
```
size_t strlen(const char *p)
{
    char *k = p;
    while(*p != NULL)
    {
        p++;
    }
    return p - k;
}
```

$105 - 100$   
char add

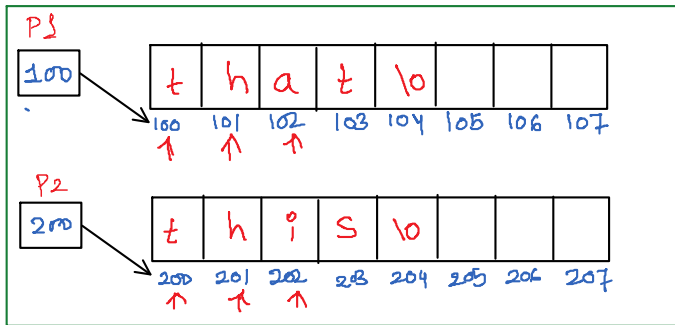
$p = p + 1$   $100 + 1 = 101$   
 $p = p + 1$   $101 + 1 = 102$   
 $p = p + 1$   $102 + 1 = 103$

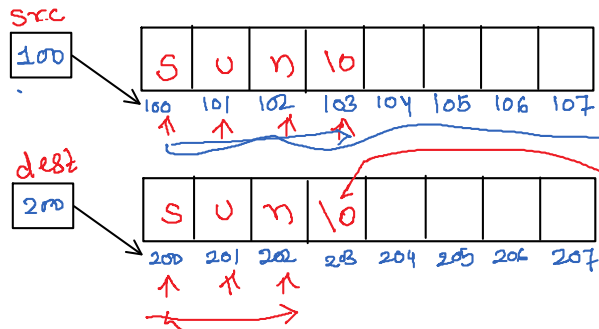


```
int mystrcmp(const char *p1, const char *p2)
{
    while(*p1 != NULL && *p2 != NULL)
    {
        if(*p1 != *p2)
            break;
        p1++;
        p2++;
    }
    return *p1 - *p2; // 'o' - 'o' = 0 - 0 = 0
}
```



— Ascii value difference





```
char *strcpy(char *dest, char *src)
{
    char *k = dest;
    while(*src != NULL)
    {
        *dest = *src;
        dest++;
        src++;
    }
    *dest = '\0';
    return k;
}
```

```
char *strcpy(char *dest, char *src)
{
    char *k = dest;
    int i=0;
    While(src[i]!=NUNLL)
    {
        dest[i] = src[i];
        i++;
    }
    dest[i] = '\0'
    return k;
}
```