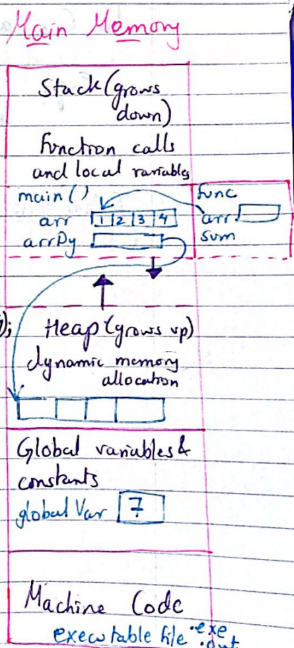
APS 105 Lecture 22

Last time: 2D arrays and how to dynamically allocate them

Today: Big picture recap and strings

Example code

```
#include <stdio.h>
#include <stdlib.h>
int globalVar = 7;

int func (int [], int);

int main (void){

    int arr [] = {1, 2, 3, 4};
    int *arrDy = (int*) malloc (sizeof(int)*4);
    printf("Sum of arr elements %d",
                func (arr, 4));

    return 0;

}

int func (int arr [], int Size){
    int sum = 0;
    for (int i=0; i<Size; i++)
        sum += arr[i];

    return sum;
}
```



Do we have variable that store words/sentences/phrases?

Strings

Did we see strings before? Yes

E.g. `printf("Hello world\n");`

"Hello world\n" is a string

In C, a string is a null-terminated array of chars.

E.g. `char myString[10];`

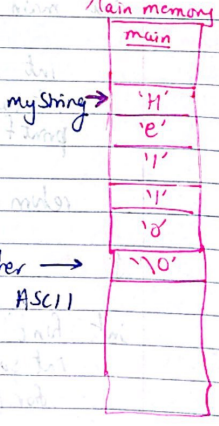
for myString to be a string, it should hold 9 characters and last element as '\0'

Initialization:

`char myString[] = "Hello";`

or

`Char myString[6] = "Hello";`



null is a character type '\0' with ASCII of zero

```
char myString[8] = "Hello";
```

'H'	'e'	'l'	'l'	'o'	'\0'	'\0'	'\0'

We declare and initialize separately:

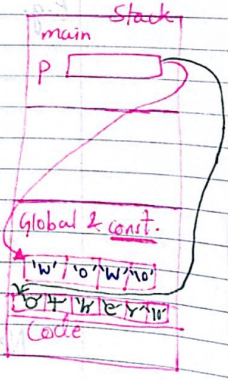
```
char s[7];
s[0] = 'T';
s[1] = 'h';
s[2] = 'e';
s[3] = '\0';
```

Note that a char uses single quote
But strings use double quotes

There is another way to declare and initialize strings (may cause confusion, so **ATTENTION!**)

```
char *p = "Wow";
```

will point to the 1st element in the array of characters



CAN DO

```
p = "Other";
```

will make p point to another string array

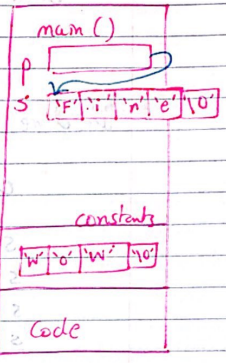
CANNOT DO

```
p[0] = 'O';
```

"Wow" is considered a constant, so you can't change individual elements → Run-time error ←

But if we have p not pointing to const Main memory.

```
char *p = "Wow";
char s[] = "fine";
p = s; *
```



Then, you can do

```
p[0] = 's';
s[1] = 'q';
```

you cannot do

```
s = "Hello";
```

s is an address not pointer value, so you can't change it
→ COMPILE-TIME error"

E.g.

```
int spaceCounter (char *s) {        or char s[]

    int count = 0;
    for (int i=0; s[i] != '\0'; i++)
        if (s[i] == ' ') count++;

    return count;
}
```

Next time Strings I/O