Passing Arrays as parameters

type name[];

type *name;

typedef unsigned long size_t;

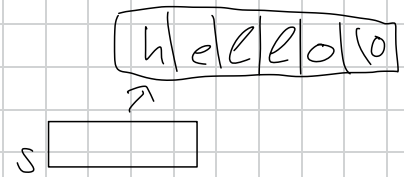
#include <cs1istd.h> we are not going to mess with it

```
size_t strlen(const char s[]){
    size_t len;
    for (len = 0; s[len] != '\0'; len++)
        /*whee*/;
    return len;
}
```

or just s[len] bc \0 = false

or

```
size_t strlen(const char *s){
    size_t len = 0;
    while (*(s++))
        len++;
    return len;
}
```



```
void *malloc(size_t size);
```

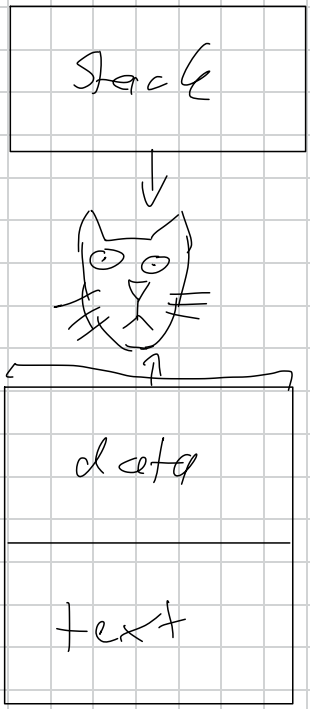
CHECK FOR ERROR

```
void free(void *victim);
```

```
void *calloc(size_t nmemb, size_t size);
```

nmemb: number of elements; size: size of element

```
void *realloc(void *ptr, size_t size);
```



e.g. $ valgrind ./detab

valgrind Memory analyzer

```
char *strdup(const char *s){
    char *new;
    if (new = malloc(strlen(s)+1))
    {
        strcpy(new, s);
    }
    return new;
}
```

NULL Byte '\0'

```
int *A;
A = malloc(10*sizeof(int))
for(i=0; i<20; i++)   <- Over Boundaries
    A[i] = 3;
```