

## Lawrence's Angrave - CS241-02 See-C-Crash

Today: Macros | Pointer arithmetic | printf buffer | malloc/free

Q0. Fix my malloc implementation

```
size_t total= 0;
char space[10000];
void* malloc(size_t request) {
    if(request + total > _____) return NULL;
    total += request;
    return _____;
}
void free(void* ptr) { ____?____ }
```

Q1. How/why does this macro work?

```
#define numelements(A) sizeof(A) / sizeof(A[0])
```

```
int data [] = { 10,20,30};
for(int i = 0; i < numelements(data); i++ ) { ... }
```

Q2. Why is this code broken?

```
#define max(a,b) a<b ? a : b
int result = max(10,5) + 1;
printf("Result:%d", result);
```

Q3 Spot the error(s)

```
int fl(int n) {
    int i;
    double* data = malloc( n* sizeof (double*) );
    while( i < n) data[i++] = 12.3;
}
```

Q4 Is the following line valid?

```
printf("%p %p", main, malloc);
```

Q5 What do you think of my logging function? How would you fix it?

```
char buffer[10000];
void log(const char*why, const char*mesg) {
    strcat(buffer, why);
    strcat(buffer, mesg); // append to my internal buffer

    if( strlen(buffer) < 500) return; // Do no writing for now
    write(2, buffer, strlen(buffer)); // Send all of the buffer to stderr
    *buffer = 0; // ?
}
```

Q6 When will printf call write(1, buffer,...) ?

Q7 Pointer arithmetic

Write a function to return the number of items in an int array before a value of -1 is found. Tricky: Use pointer arithmetic (no counters allowed!)

```
count_before(int* array) {
    int* ptr = array;
```

Q8 What would you call at line 2 such that p1 can be equal to p2?

```
void* p1 = malloc( 10 );
??_____
```

```
void *p2 = malloc(8);
```

Q9 When are `sprintf` and `fprintf` useful?

Q10 `main(int argc, char** argv)`

What is special about `argv[0]`

What is special about `argv[argc]`

How do you print out all of the arguments of a program?

```
int main(int argc, char** argv) {
```

Q11 Which of the following would print out an address in the stack?

```
int abc = 5;  
int main() { fl( 10, abc); return 0;}
```

```
int fl(int v1, int* v2) {  
    printf("%p", &v1);  
  
    printf("%p", &v2);  
  
    printf("%p", v2);  
}
```

Q12 Which of the following is incorrect? Can you fix it?

```
const char* f1() {  
    const char blah[] = "Hello";  
    return blah;  
}
```

```
const char* f2() {  
    const char* foo = "Hi";  
    return foo;  
}
```

```
const char* f3() {  
    const char* yo = malloc(2);  
    strcat(yo, "Hi");  
    return yo;  
}
```

Q13

How do I change your variable? Complete the main function to print the message created by the getMessage function

```
int main() {  
    _____  
    puts(_____);  
}
```

```
void getMessage(char** mesg) {  
    *mesg = malloc(10);  
    strcpy(*mesg, "I C!");  
}
```