
# Midterm Review

# COS 217 Midterm: Wed Mar 11


## When/where?

- In lecture, W 3/11; *Friend 101* (P0{1,2,3,5,6}) and *CS 105* (P04)

## What?

- C programming, including string and stdio features we've seen
- Numeric representations corresponding to C types we've seen
- Programming in the large: modularity, building, testing, debugging
- Readings, lectures, precepts: through *this week*
- Assignments 0-3.

## How?

- Mostly short-answer, focused on code reading. Some code writing.
- Closed book and notes
- No electronic anything
- Interfaces of relevant functions will be provided

## Old exams are posted on schedule page

# After that …

## When/where?

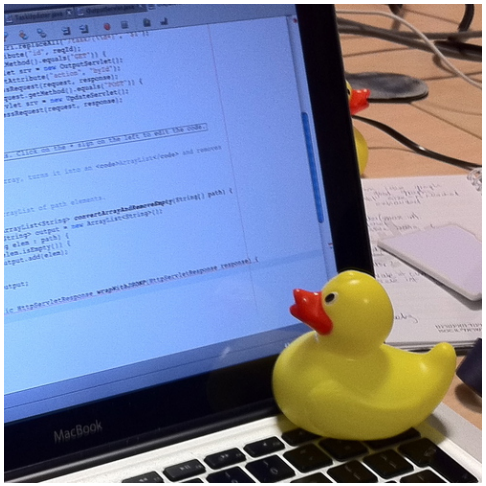
- ???

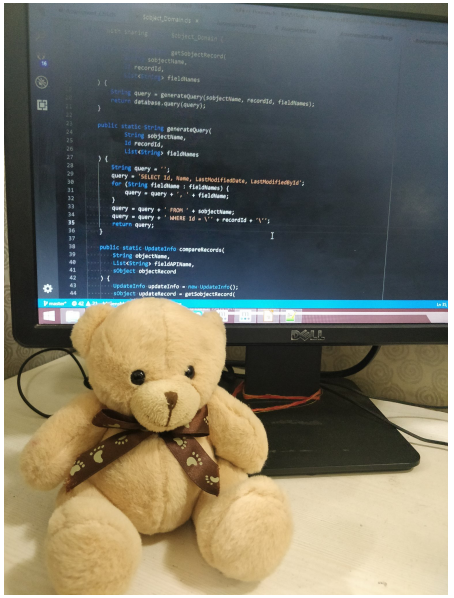
## What?

- Same as planned.

## How?

- ???





## We'll do our best to make things work.

- Don't mean to (rubber) duck (debugging) the question, we just don't know yet.
- (Teddy) Bear (debugging) with us.

# Fall 2015, Question 1d


What does `printf("%d", (0532 << 3)/64)` print to stdout?

- << 3 shifts *left* by three bits
- /64 is dividing by $2^6$, or shifting *right* by six bits
- There are 0's in the original leftmost 4 bits
- So, the result is shifting right by three bits

- 0532 is an octal number (three bits per octal digit)
- Shifting right by three bits is 053
- Converting to decimal ("%d") is 5*8 + 3*1, or 43

# Spring 2015, Question 1


Indicate which of these expressions evaluates to true or false

| | |
|---|---|
| 1a) ~1 && 1 | TRUE (note *logical* AND) |
| 1b) 512 - 01000 | FALSE (note *octal* number) |
| 1c) 0x2B \| (! 0x2B) | TRUE (note *logical* NOT) |
| 1d) 16 >> 4 | TRUE ($10000_2$ >> 4 is 1) |
| 1e) sizeof(5) > sizeof(2L) | FALSE (int/4 vs long/8) |
| 1f) -10 < i < -1 | FALSE (left-to-right associativity) |

# Fall 2012, Question IV(b)


What is the risk with having the following line of code in a C program:

```
assert (f() == 0);
```

How would you rewrite the code to eliminate the risk?

- Assert statements can be disabled in compilation (NDEBUG macro)

- We do not know what calling f() might do
- E.g., side effects like changing a global variable

- Instead, do

```
int i;
i = f();
assert(i == 0);
```

# Spring 2014, Question 2(d)


Assuming y is valid, what would you put in the body of the function if the function is to return zero if and only if first parameter is equal to the integer value accessible through the second parameter.

```
struct z {int x;};

typedef struct z z;

static int Func(int x, z *y) {

   return y->x != x;

}
```

# Spring 14, Question 3


What does this code do? (In parts)

```
int main(int argc, char *argv[]) {
    char *a = argv[1]; /* gets first arg */
    char *b;
    int k, i, tuk;

    assert(a != NULL);
    for (k = 0; a[k] != '\0'; k++)
        ;
```

Computes length of string and stores in k

# Spring 14, Question 3


What does this code do? (In parts)

```
assert(a != NULL);
for (k = 0; a[k] != '\0'; k++)
   ;

tuk = k<<1;
b = malloc(tuk + 1);
b[tuk] = '\0';
```

Allocates space for string twice the size, and terminates the string

# Spring 14, Question 3


What does this code do? (In parts)

```
assert(a != NULL);
for (k = 0; a[k] != '\0'; k++)
  ;

tuk = k<<1;
b = malloc(tuk + 1);
b[tuk] = '\0';

for (i = 0; i < k; i++)
   b[i] = a[i];
```

Copies the original string in the first half of the new space

# Spring 14, Question 3


What does this code do? (In parts)

```
tuk = k<<1;
b = malloc(tuk + 1);
b[tuk] = '\0';

for (i = 0; i < k; i++)
    b[i] = a[i];

for (i = 0; i < k; i++)
    b[i+k] = a[k-1-i];
```

Copies reversed version of the string into the second half of the new space

# Spring 14, Question 3


What does this code do? (In parts)

```
tuk = k<<1;
b = malloc(tuk + 1);
b[tuk] = '\0';

for (i = 0; i < k; i++)
   b[i] = a[i];

for (i = 0; i < k; i++)
   b[i+k] = a[k-1-i];

printf("%s\n", b);
```

In total: creates a palindrome and prints it!

# Fall 2015, Question 2(b)


This function should print every other character of the input (i.e., for an input of "0123...", the output should be "13..."). When does it produce the wrong answer? Rewrite the code to fix the bug.

```
void q2b(void) {
    while (getchar() != EOF)
        putchar(getchar());
}
```

Wrong when number of characters is odd.

# Fall 2015, Question 2(b)


Rewrite the code to fix the bug.

```
void q2b(void) {
   int c;
   for (;;) {
      c = getchar();
      if (c == EOF)
        return;
      c = getchar();
      if (c == EOF)
        return;
      putchar(c);
   }
}
```

# Fall 2015, Question 2(c)


This function should return the maximum value in an array a of `n` integers. When does this code return the wrong value? Modify the code to correct the bug.

```
int q2c(int *a, int n) {
    int currmax = 0, i;
    assert(a != NULL);
    assert(n > 0);
    for (i = 0; i < n; i++)
        if (a[i] > currmax)
            currmax = a[i];
    return currmax;
}
```

Negative numbers!

# Fall 2015, Question 2(c)


This function should return the maximum value in an array a of `n` integers. When does this code return the wrong value? Modify the code to correct the bug.

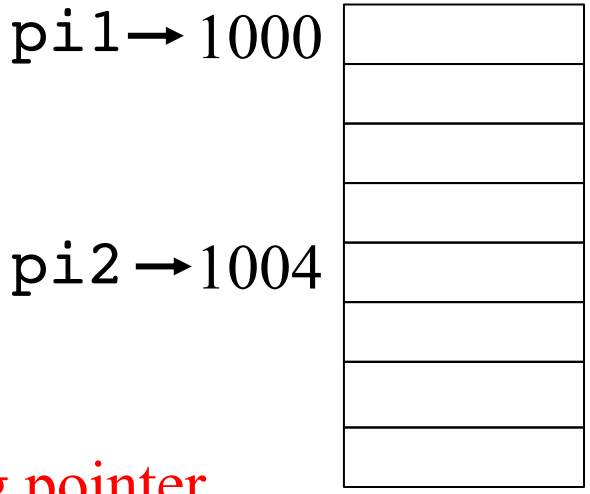
```
int q2c(int *a, int n) {
    int currmax, i;
    assert(a != NULL);
    assert(n > 0);

    currmax = a[0];
    for (i = 1; i < n; i++)
        if (a[i] > currmax)
            currmax = a[i];
    return currmax;
}
```

# Spring 2018, Question 4


Assume that each of the following snippets of code is run with these variables defined:

```
int *pi1 = (int *) malloc(2 * sizeof(int));
int *pi2 = pi1 + 1;
int i = 1;
```




What memory management error(s) result from each code snippet?

```
free(pi1);
*(pi2 - 1) = i;
free(pi2);
```

Dangling pointer

Free of unallocated memory

# Spring 2018, Question 4

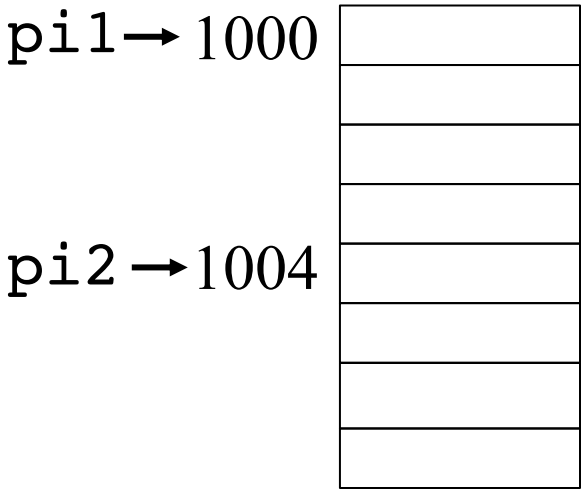

Assume that each of the following snippets of code is run with these variables defined:

```
int *pi1 = (int *) malloc(2 * sizeof(int));
int *pi2 = pi1 + 1;
int i = 1;
```

What memory management error(s) result from each code snippet?

```
free(pi1);
pi2 -= 1;
free(pi2);
```

Double free



# Fall 2012, Question 1(g)


You have written a module to implement a queue, and published an interface for it in a .h file. A colleague reviews it and suggests you add a function to the interface. What are the two questions you ask yourself to decide whether or not to include the new function in the module interface?

Is the function necessary to make the module complete?

Is the function convenient for many clients?


See you on Wednesday!

( And then … who knows? :/ )