$CS241 #02 *See C Crash !*

1. Test your neighbor.
What is the C preprocessor? Give 3 examples of its use in a .c file
When is sizeof evaluated?
What would be a good name for the following macro? What must be true for A?

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

2. Complete the following code to return 1 if the c string contains @ character,0 otherwise.
Where will the code crash if called with NULL containsAt(NULL);

```
01  int containsAt(char* ptr) {
02
03
04
05
06
07
08
```

3. Explain how C uses memory from the process address space in each line of the following. (eg. stack, text segment, heap, global).
Where do you expect this code to fail?

```
01  int global;
02
03  void test() {
04    char* t1 = "hi"; // Init a pointer
05    char t2[] = "abcdefgh"; // copies bytes
06
07   *t2 = 'A';
08   *(t2 + 1) = 'B';
09   t2[1] = 'B';
10   *t1 ='H';
11   t1 = malloc(123);
12  }
```

4. Why is this code broken?

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

5. Spot the error(s)

```
01  double* f1(int n) {
02   int i;
03   double* r = malloc( n* sizeof (double*) );
04   while( i < n) r[i++] = 12.3;
05   return r;
06  }
```

6. Is the following line valid?

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

7. 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!)

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

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

```
01  void* p1 = malloc( 10 );

02  ??_____

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

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

12 Examples of using `strcpy, strcat, strlen, strcmp`

10. For the start of the program, `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?

```
01  int main(int argc, char** argv) {
02
```

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

```
01  int abc = 5;
02
03  int main() { f1( 10, &abc); return 0;}
04
05  int f1(int v1, int* v2) {
06     printf("%p", &v1);
07
08     printf("%p", &v2);
09
10     printf("%p", v2);
11  }
```

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

```
01  const char* f1() {
02    const char blah[] = "Hello";
03    return blah;
04  }
05
06  const char* f2() {
07    const char* foo = "Hi";
08    return foo;
09  }
10
11  const char* f3() {
12    const char* yo = malloc(2);
13    strcat(yo, "Hi");
14    return yo;
15  }
```

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

```
01  int main() {
02    ?
03
04    puts( _____);
05  }
06  void getMessage(char**mesg) {
07    *mesg = malloc(10);
08    strcpy(*mesg, "I C!");
09  }
```