CSci 2113 - Lecture 3: C Data Structures

Prof. T. Wood Fall 2016

1. Fill in the contents of the stack and heap up to the marked line. Assume `int` and `int*` use 4 bytes and no extra space is needed for function headers on the stack or heap meta data. Assume heap objects are allocated into the first free space of sufficient size.
The stack starts at 10,000 and grows up, while the Heap starts at 50,000 and grows down.

```
void firstFunc() {
  int a = 20;
  int b = 30;
  int *p = malloc(sizeof(int));
  *p = 5;
  b = *p;
  free(p);
  secondFunc();
  p = malloc(sizeof(int));
  *p = 50;
  /* DRAW MEMORY @ THIS LINE */
}
void secondFunc() {
  int *q = (int*) malloc(sizeof(int));
  int *r = (int*) malloc(sizeof(int));

  *q = 45;
  *r = 100;
  free(q);
  return;
}
int main() {
  int x = 10;
  int *y = &x;
  firstFunc();
  return 0;
}
```

Stack		
Address	*Name*	*Contents*
10000		
10004		
10008		
10012		
10016		
10020		
10024		
10028		
10032		

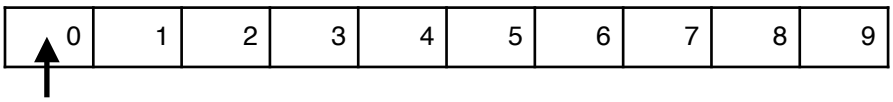
Heap		
Address	*Name*	*Contents*
50000		
49996		
49992		
49988		
49984		
49980		
49976		
49972		
49968		

Self-Quiz

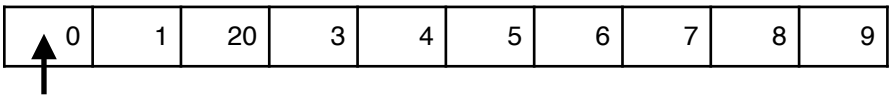
1. Suppose an int consumes 4 bytes and a char is 1 byte. If you add 1 to an int pointer, what does that do to its address? What about adding one to a char pointer?
2. What is the benefit of a char*[] compared to a char[][]?
3. The notation for using a struct is....
4. What are the differences between an array and a linked list? When is one faster than the other?
5. How much memory would an int array with 10 elements consume? What about a linked list that stores one int per entry? (explain any assumptions you make)

2. After each line of the program, indicate the contents of the full array and use an arrow to indicate the entry pointed at by the **`array`** pointer.

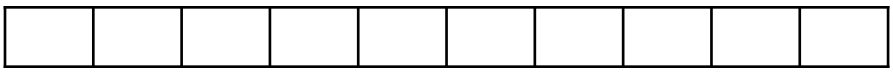
```
int* array = (int*) malloc(sizeof(int)*10);

// fill array with the values 0...9
```



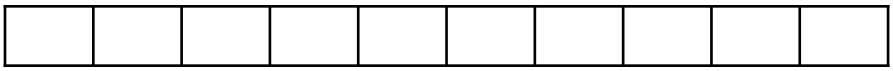

```
array[2] = 20;
```



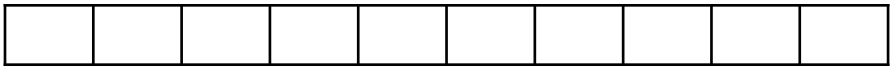

```
*array = 100;
```



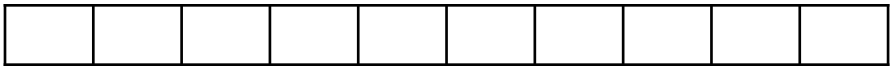
```
array += 1;
```



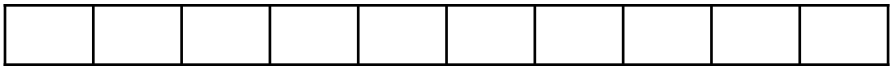
```
*array = 2;
```



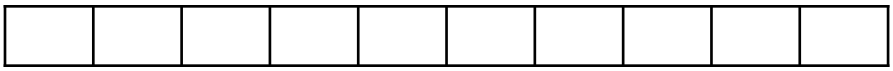
```
array[3] = 60;
```



```
array = array + 2;
```



```
array[0] = array[0] + 2;
```



```
*(array + 3) = 90;
```