Exercise 4

EE312 Fall 2013

To complete during recitation 9/26 and 9/27

TWO POINTS

NOTE: I've left all the type casts off for the code examples/questions below. We do not consider type casts to be materially significant to your understanding of pointers. Pointer types **are** important, *casting* one type of pointer to another is (mostly) style and not substance.

1. Write a function that searches the heap for the closest fitting available chunk for the specified size. Specifically return the index within the heap of the bottom signature for the smallest available chunk that has at least *size* words (a word is 4 bytes, and is the amount of memory required to hold an **int** variable). You can assume that the heap is maintained in a global variable (an array of ints) called *heap*. The number of elements in this array is *heap_size*. Do not change the heap in any way, simply identify and return the best fitting chunk currently in the heap. You should assume that there is at least one chunk large enough to satisfy the request.

```
const int heap_size = …; // some value
int heap[heap_size];

// return the index (not the address) of the bottom signature of the chunk
int bestFit(int size) {
```

2. For each program, draw a diagram of the heap and answer the question(s) posed. You should assume that the heap is an array of 10 **int** locations beginning at address 1000. The following diagram shows the initial state of the heap. You should assume that each of the following questions starts from this state. The "?" indicates that the value stored in that memory location is unknown. Please show only the final state of the heap. If you need scratch paper, please write on the back of the page **WE NEED TO BE ABLE TO READ YOUR DIAGRAM**. No credit will be given for illegible answers. Be sure to both draw the diagram of the heap and answer the question. You must include both an answer and a diagram to receive credit. REMEMBER the argument to malloc is in units of BYTES

the heap

address	contents
1036	8
1034	?
1028	?
1024	?
1020	?
1016	?
1012	?
1008	?
1004	?
1000	8

a. (6 pts)

```
int* p = malloc(1);
int k;
for (k = 0; k < 5; k = k + 1) {
  p[k] = -1;
}
for (k = 5; k < 9; k = k + 1) {
  p[k] = 2;
}
int* q = malloc(1);
```

What address is stored in the variable q?

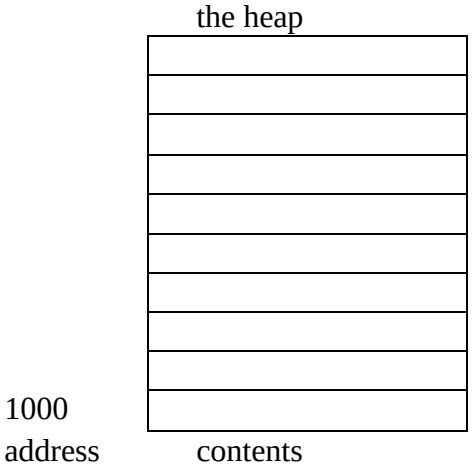
the heap

address	contents
1000	

b. (6 pts)

```
int* p = malloc(1);
*p = 42;
int* q = malloc(1);
free(p);
int* r = malloc(1);
*q = *r;
```

How large (in bytes) is the largest chunk still available?



c. (4 pts) (arg to malloc is bytes!)

```
int* p = malloc(6);
int* q = malloc(4);
int* r = malloc(4);
free(r);
free(q);
free(p);
```