First name (color-in initial)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (9,10,11, 12,1 or 2)	first name initial	last name initial
Last name (color-in initial)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

H12: Due Tuesday, 02.10 in Lab

Structs (Sect 10.1), Linked Lists with Structs (Sect. 13.1 up to p. 743 ONLY))
Assigned: Mon 02.02 Total Points: 50

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE,
or offered in person, for in person grading, during instructor or TAs office hours.
See the course syllabus at https://foo.cs.ucsb.edu/16wiki/index.php/W15:Syllabus for more details.

---

(1) (10 pts) Fill in the information below. Also, fill in the A-Z header by

- **coloring in** the first letter of your first and last name (as it appears in Gauchospace),
- writing **either 9,10,11,12,1 or 2** to indicate your **discussion section (lab)** meeting time
- writing your **first and last initial** in large capital letters.

All of this helps us to manage the avalanche of paper that results from the daily homework.

name:	
umail address:	@umail.ucsb.edu

If you collaborated with AT MOST one other person on this homework, write his/her name below. She/he should also have your name on his/her paper.

Reading: *Read Sections 9.1 (pointers), 10.1 (structs) and 13.1 (up to p. 743 ONLY, the part on linked lists with structs.)*

Then, answer the following questions. Be sure to check both sides.

1. (5 pts) Suppose you need to track information about dates in a program. Write the C++ code for a `struct Date` (note the capital D) that contains three data members of type `int` called m, d and y. (While we normally avoid single letter variable names, a case can be made that here the meaning will be clear.)

2. (5 pts) Suppose you are writing a program to track data about earthquakes. Write the C++ code for a struct definition that represents information about a single earthquake. The name of the struct should be Earthquake (with a capital E). It should contain three data members: latitude (a double), longitude (a double) and date. Use your struct Date from the previous problem as the type of the date member. Then, include an `Earthquake *next;` data member so that the struct can be used to create a linked list of Earthquakes.

All of the questions on this page pertain to the program listed at the right of this page.

4. (18 pts) Fill in the following table, indicating the type of each expression. The first two are done for you as examples.

expression	type
`x`	int
`&x`	int *
`p`	
`*p`	
`y + x`	
`p->next`	
`x + 1`	
`p->data`	
`&y`	
`*(p->next)`	
`b`	
`head`	
`*a`	
`head->next->next`	
`&a`	
`head->next->data`	
`n`	
`argv[0]`	
`&n`	
`argc`	

5. (12 pts) For each of the following, circle whether the item indicated can be found on the stack or the heap (as of the time that the return 0; is reached.)

`x`	stack heap
`b`	stack heap
`*a`	stack heap
`n`	stack heap
`head`	stack heap
`*head`	stack heap

```
#include <iostream>
struct Node {
  int data;
  Node *next;
};

int main(int argc, char *argv[]) {
  int x;   double y;
  int z;   int *a;
  double *b;

  Node n;
  n.data = 12;
  n.next = NULL;

  x = 3;    y = 7.3;
  z = 8;
  a = &z;  b = &y;

  Node *head = NULL;
  Node *tail = NULL;
  Node *p = NULL;

  head = new Node;
  head->data = 3;
  head->next = NULL;
  tail = head;

  p = new Node;
  p->data = 6;
  p->next = NULL;
  tail -> next = p;
  tail = p;

  p = new Node;
  p->data = 10;
  p->next = NULL;
  tail -> next = p;
  tail = p;

  return 0;
} // end main
```

HINTS: As a reminder: storage that is allocated with **new** is on the **heap**, what our textbook calls the "freestore". Local variables (declared inside the braces of a function definition, including main), and parameters to functions (including the argc and argv parameters to main) are called "automatic variables" by our textbook, and are stored on the stack. Keep in mind: it is very common for a *pointer on the stack* to point to *storage allocated on the heap*. So be clear about whether the question is asking about the pointer, or the pointee!