

Name: <i>(as it would appear on official course roster)</i>		section
Umail address: _____@umail.ucsb.edu		
Optional: name you wish to be called if different from name above.		
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")		

1

h06

CS16 S18

h06: Chapter 7 and 10: Pointers

ready?	assigned	due	points
true	Fri 05/04 09:00AM	Fri 05/11 11:59PM	53

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE, OR IF APPLICABLE, SUBMITTED ON GRADESCOPE. There is NO MAKEUP for missed assignments; in place of that, we drop the three lowest scores (if you have zeros, those are the three lowest scores.)

Read Chapter 7.1 - 7.2 up to page 397. Also read this handout on Pointers and Memory: <http://cslibrary.stanford.edu/102/PointersAndMemory.pdf> You don't need to turn this homework in. To earn credit for this homework, complete the corresponding quiz on gauchospace AFTER you have completed the pen and pencil version of the homework. The quiz will be available one day before the due date indicated on the homework.

**PLEASE MARK YOUR HOMEWORK CLEARLY, REGARDLESS OF IF YOU WRITE IT OUT IN INK OR PENCIL!
FOR BEST RESULTS, PRINT THIS PAGE AS A PDF, THEN PRINT THE PDF**

Please:

- No Staples.
- No Paperclips.
- No folded down corners.

1.(5 pts) What is the output of the following code? If there's an error that will not allow an output, point it out. Briefly justify your answer.

```
int arr[5];
for (int i = 0; i < 5; i++) {
    if (i < 3) arr[i] = 'a';
    else arr[i] = 'z';
    cout << arr[i] << endl; }

```

ascii value of 'a'] 3 times
 " " " "]
 " " " "]
 ascii value of 'z'] 2 times
 " " " "]

2.(5 pts) What is the output of the following code? If there's an error that will not allow an output, point it out.

```
int arr[7] = {5};
for (int i = 0; i < 7; i++)
    cout << arr[i] + i << " ";

```

5 1 2 3 4 5 6

3.(5 pts) What is the output of the following code? If there's an error that will not allow an output, point it out.

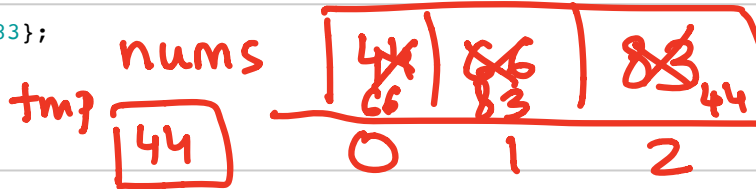
```
int codes[] = {44, 66, 83, 973, -977};
for (int count : codes) {
    if ( (count/2) < 50 )
        cout << count << endl;
    else cout << "invalid" << endl; }

```

44
 66
 83
 invalid
 -977

4.(5 pts) Draw a diagram to show how the content of the array 'nums' changes in memory after every line of the following code is executed. Start by showing the elements of the array in memory when the array is initialized. Every time the element of an array changes, you may indicate the change by crossing out the old value and writing in the new value.

```
int nums[] = {44, 66, 83};
int tmp = nums[0];
nums[0] = nums[1];
nums[1] = nums[2];
nums[2] = tmp;
```



5.(5 pts) Describe in your own words what the code in the previous question does. Your description of the code should be as abstract as possible. For example.. "the above code sorts the elements of the array in ascending order..."

Rotates the elements left

2

h06

CS16 S18

6.(10 pts) Write the definition of a function named 'reverse' that takes two parameters: an integer array and the number of elements of the array. The function should reverse the order of elements of the array. The function should not return anything. See below for an example of expected outcome when the reverse function is called:

```
int nums[] = {10,20,30,40,50};
reverse(nums, 5); // The order of elements should be reversed after this
for(int i=0; i<5; i++)
    cout<< nums[i]<<" "
```

The output of the above code should be: 50 40 30 20 10

```
void reverse ( int* arr , int len ) {
    int tmp;
    for (int i=0 ; i < len/2 ; i++) {
        // swap arr[i] & arr [ len-i-1 ]
        tmp = arr[i]
        arr[i] = arr [ len-i-1 ]
        arr [ len-i-1 ] = tmp;
    }
}
```

3

7.(4 pts) List two reasons why you might choose to pass parameters to a function by address or reference, rather than by value in your C++ programs.

- More efficient
- Allows modification of data that is outside the scope of the function

8.(6 pts) Draw a pointer diagram to demonstrate how the state of memory changes as the following code is executed. Cross out old values/arrows and draw new ones. Is this program likely to result in a segmentation fault? If so, why?

```
int num = 10, *ptr1 = &num, *ptr2=0;
if (ptr2) ptr1 = ptr2;
else if(ptr1) ptr2 = ptr1;
(*ptr1)++;
```

h06

CS16 S18

9.(8 pts) On page 14 of the “Pointers and Memory” handout there is a reference to the “Amperand (&)” Bug in the TAB() function. Draw a trace of the local variables of the functions Victim() and TAB() on the run-time stack as these functions are executed. Your diagram should be similar to the one provided on page 13. Your diagram should depict exactly three instances in program execution - T1: Right after the first statement of Victim(), T2: right before the return statement in TAB(), T3: Right after the call to TAB(). Use your diagram to explain the “Amperand (&)” bug and why it is likely to lead to a runtime error when the last line of the Victim() is executed?

T1 - Victim()'s locals have been allocated and given values	T2 - TAB() is called, and its locals are allocated.	T3 - TAB() exits and its locals are deallocated. Victim()'s locals will be deallocated when it exits.
Victim() ptr 	Victim() ptr ↓ TAB() temp 	Victim() ptr ↓ 42

4

h06

CS16 S18