CS 202 fall 2017, exam 1 on paper part

cs202 login: _____

Name: _____

Each question is 1 point and is graded as right, wrong, or half credit. 43 points total.

For stack questions, delete means pop (and ignore parameter #) and lookup means top. For queue questions, delete means dequeue (and ignore parameter #) and lookup means front For the following sequence of operations, write what each data structure would look like at the end. Insert 50, insert 40, insert 60, insert 70, insert 80, insert 20, insert 10, insert 30, delete 10, delete 40, insert 90

stack

queue

BST

unsorted array

sorted array

hash table (with hash function h(x) = x % 9)

Complete the following table to give the "big O expression" for the <u>worst-case</u> running times. For hash table, put the "<u>normal case</u>" running times. *(1 point per row)*

	lookup	insert	delete (after lookup is already done)
stack			
queue			
unsorted array			
sorted array			
BST (for balanced tree)			
hash table			

Complete the following for using gcc on CS. 1 point per row.

	# bytes	Min value	Max value
char			
unsigned short int			
int			
unsigned long int			
float		NA	NA
double		NA	NA
unsigned char			

Give an extremely short phrase (at most 10 words) describing what each keyword does. 1 point each.

const			
static			
if			
switch			
return			

For each, give	the value(s)	printed. 2	points each.
----------------	--------------	------------	--------------

printf("%c", (96 >> 3) << 2);	
printf("%i, %i, %f\n", 10/3, 10 % 3, 10/3);	
int x = 1; if ((x = 0) && (x++)) x++; printf("%i", x);	
int x = 35; while (x > 0) { printf("%i", x % 3); x /= 3; }	
int x[5] = {5, 6, 7, 8, 9}; int * y = &x *x /= 3; y++; y[1] = 3; int * z = y++; printf("%i, %i, %i", x[0], *y, *z);	

Play computer. Keep track of what happens to each variable. 3 points. If you have the final values of the variables in main correct you get full credit.

```
int fun1(char *s) {
 if (s[0] == (0) return 0;
 char *next = & (s[1]);
 if (s[0] == a) return fun1(next) + 1;
 else if (s[0] == b) return fun1(next) + 2;
 else return fun1(next) + 0;
}
int fun2(char *s) {
 if (s[0] == (0) return 0;
 for(int i=0; s[i] != '\0'; i++)
  s[i] ++;
return fun1(s);
}
int main(int argc, char *argv[]) {
 char s1[10] = "aaabbb",
  S2[10] = "ababab";
 int x = fun1(s1);
 int y = fun2(s2);
 x = y;
 y = x;
 return 0;
}
```

Write a C program. 3 points each

Write a complete C program (including #include and everything) that: prints how many command-line arguments were given, and prints them in reverse order. For example, running ./a.out hello there would print

arguments: 3 there hello

./a.out

Write the main function for a C program that reads an integer, reads that many integers, and prints the average of the odd #'s. For example, running ./a.out and then typing the values 5, 2, 4, 7 would print Average of odds: 6.0