CSE 3100 Exam 1 Rubric- Fall23

Problem 1- (max 30 points)		
maximum possible points through partial grading - 27 points		
Criteria		**Points**
letter_counts()		Total of 20
Initializing counts[]	looping through all indices of counts[]	5
	assigning counts[i] =0	5
calculating count for each letter	Looping thorugh all n letters	5
	incrementing/calculating count of each letter	5
main()		Total of 7
Printing the output	Looping through all letters correctly	2
	checking if count of a letter is >0	2
	using printf()	1
	correct string literal - "%c %d\n"	1
	using correct variables to print	1

Problem 2-(max 40 points)			
maximum possible points through partial grading - 36 points			
Criteria			**Points**
match() from match.c			Total of 25
Pushing bracket to stack in the first if{} block in for loop	calling push() function		2
	stack s as first parameter in push()		1
	create_node() as second parameter in push()		1
	exp[i] as parameter in create_node() above		1
else if {} block in for() loop checking if stack is empty and popping nodes if stack is not empty	breaking if stack is empty	checking if stack is empty	1
		assigning result=0 if stack is empty	1
		break; if stack is empty	1
	popping bracket from stack	calling pop(s)	2
		freeing the popped node using free()	1
Checking if current bracket matches with bracket in node popped (a different implementation will still lead to a total of 12 points)	if '(' and ')' didn't match	checking if exp[i]=') and popped element is not '('	2
		assigning result=0 if not matched	1
		break; if not matched	1
	if ']' and '[' didn't match , same point division as above		4
	if '}' and '{' didn't match , same point division as above		4
if(!empty(s)) {}block	assigning result = 0		1
	calling clear_stack(s)		1
push() from stack.c			Total of 3
pushing element to stack	call add_first()		1
	&s->top as first parameter		1
	new_node as second parameter		1
pop() from stack.c			Total of 3
popping element to stack	calling remove_first()		1
	&s->top as first paramter		1
	return the popped node		1
empty() from stack.c			Total of 2
checking if stack is empty	checking if s->top == NULL		1
	returning the above result		1
clear_stack() from stack.c			Total of 3
freeing all elements from stack	looping through all elements orf stack		1
	popping /deleting the nodes		1
	freeing the popped/deleted nodes()		1
Note: different implementation of functions in stack.c will still have same maximum amount of points as above			