Session Name: 26 - Review 4 12-5-2023 12-13 PM

Date Created: 12/5/23, 11:54:49 AM **Active Participants:** 46 of 69
Average Score: 0.00% **Questions:** 2

Results by Question

1. Please do question 1 which is on the screen in class. (Short Answer)

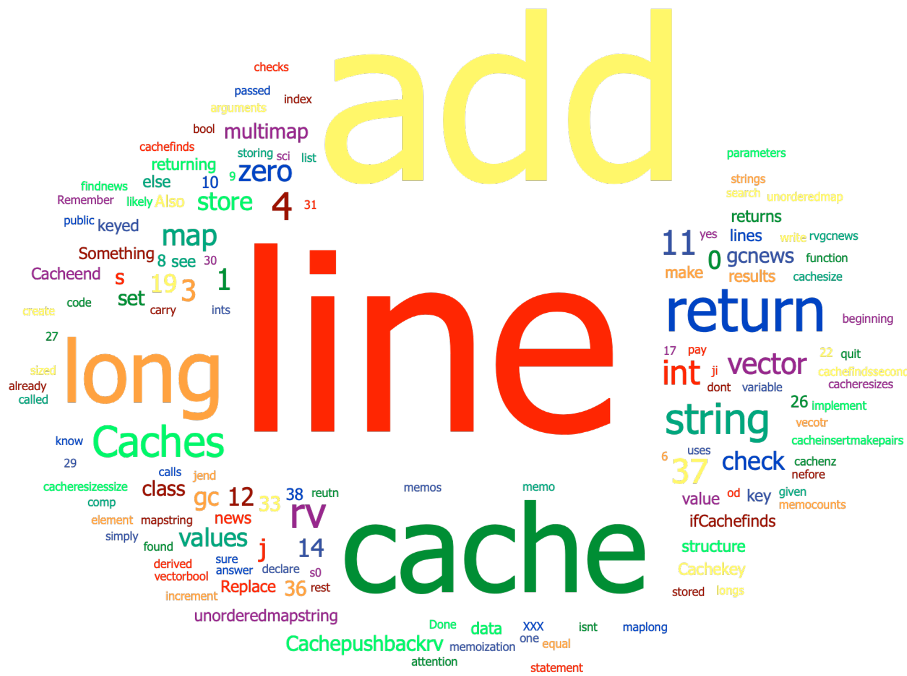
	Responses	
	Percent	**Count**
251 983 036 783 619 186 494 469 734	22.73%	10
619 983 036 783 734 186 494 469 251	9.09%	4
036,186,251,469,494,619,783,983	4.55%	2
251 036 186 494 469 619 983 783 734	4.55%	2
036 186 251 469 494 619 783 983	2.27%	1
036 186 251 469 494 734 783 983	2.27%	1
036 186 251 494 469 619 734 783 983	2.27%	1
036 186 251 494 619 734 783 983	2.27%	1
036 251 983 186 734 783 469 494 619	2.27%	1
251	2.27%	1
251 494 036 469 186 494 494 783 619	2.27%	1
251 734 619	2.27%	1
251 938 036 783 619 186 494 469 734	2.27%	1
251 983 036 734	2.27%	1
251 983 036 783 186 494 469	2.27%	1
251, 983, 036, 783, 619, 186, 494, 469, 734	2.27%	1
251,468,494	2.27%	1
251,983,036,783,619,186,494,469,734	2.27%	1
251,983,036,783,734,186,494,469	2.27%	1
494 251 036 469 186 619 783 983	2.27%	1
619 036 186 251 783 734 494 469 983	2.27%	1
619 251 036 186 494 469 983 783 734	2.27%	1
619 251 036 469 186 494 734 783 983	2.27%	1

619 251 469 494 186 734 783 036 983	2.27%	1		
619 983	2.27%	1		
619, 251, 983, 036, 783, 734, 186, 494, 469, 251	2.27%	1		
734 983 036 783 251 186 494 469 619	2.27%	1		
73425161998303678318649469	2.27%	1		
983 734 496	2.27%	1		
SORT(VEC.BEGIN(),VEC.END());	2.27%	1	**Keyword(s):**	186, 251, 036, 469, 494, 619, 734, 783, 983
Totals	**100%**	**44**	**Keyword Matches:**	0

2. Please do question 2 which is on the screen in class. (Essay)

Responses
1
11. Add cache
?
Above line 1, add //"Remember to pay attention in class"
Add a Cache data structure after line 3. Make it an unordered_map <string, long long> After line 36, add Cache[key] = gc
Add a cache data structure to the class and store values before returning above lines 19, 37. Also search the cache after line 12.
Add a cache to class XXX which is multimap keyed on strings storing values of bool. Then before the returns in gc() add element to multimap. Also add a check at the beginning of gc() that checks the multimap and returns if found.
Add cache after line 3
Add some cache here and there
After Line 1 - Add "unordered_map<string, int> Cache;" On Line 11 - Add "if(Cache.find(s) != Cache.end()) return Cache[s]; After Line 36 - Add "Cache[s] = rv ^ gc(news, zero);"
After line 10 add "map<long long> cache" After line 12 add "if (cache.size() == 0) cache.resize(s);"
After line 3, add map <string, long long> cache; At line 11, add if (cache.find(s) != cache.end) return cache[s]; At line 12, add cache[s] = 0; before return 0; At line 33, add cache[s] = rv; before return rv; At line 37, add cache[s] = rv ^ gc(news, zero); After line 37, add return cache[s] Done(???)
after line 3, add public variable list of ints from string s
At line 11 add “vector <int> cache” After that line add “cache.resize(s.size())” After line 29 add “cache[nz] = rv”
At line 27
Before 4: create map cache. Line 19: Replace with cache[s] = rv; Line 33: replace return rv with cache[key]
Before line 10, add int j (???)
Before line 11, add “set <string> j” After line 36, if find(news) == j.end()) add news to set j Else, return rv ^ news
Before line 14 add a cache that uses a key derived from ‘s’
before line 17 add cache check
before line 22, add to cache
Before line 3, add “unordered_map<string, long long> memo;” After line 11, add “if (memo.count(s) == 1) return memo[s];”
Before line 30, add memoization.

Before line 31, I would implement a check for the cache to see if a given function call's arguments have already been called. If so, simply return the results from the cache.
Before line 38, add something.
before line 4 add a vector of long long sized to 26 and set to values of 0. nefore line 14 write an if statement to see if the vecotr at index s(0) - a is not equal to zero. if it isnt then reutn the value else do the rest od the code. the store values into cache below
before Line 4 make a vector of long long before line 14 increment one Before line 26 store the string to vector for cache
Before line 4, add "vector <long long> Cache;" Before line 19, add "Cache.push_back(rv):" Before line 33, add "Cache.push_back(rv);" Before line 37, add "Cache.push_back(rv ^ gc(news zero);"
Before line 4, add map<string, long long> cache; Before line 12, add string key = s; Before line 14, add if(cache.find(s)) return cache.find(s).second; Before line 37, add cache.insert(make_pair(s, rv^gc(news, zero))
Before line 4, add vector<bool>
Before line 4, declare a cache - likely with a map. Before line 8, check if the cache has stored an answer for the passed parameters - if yes, return it, if no, carry on. Before lines 19 and 37, add the return value to the cache before returning it.
Before line 8, add “map <int, int> cache” After line 38 quit comp sci
Before line 9, add "int j"
I don’t know :(
ji
Line 1, add a cache of long longs to store results of gc
Not sure
On line 6 Add a string, long long map keyed on the string
Something


add
line
cache
long
return
Caches
string
rv
int
vector
check
map
zero
store
multimap
4
11
gcnews
returns
lines
make
results
0
37
key
value
ifCachefinds
structure
Cachekey
stored
longs
values
class
gc
12
33
38
reutn
14
j
news
Replace
36
rest
unorderedmapstring
Cachepushbackrv
memoization
memos
memo
Done
data
XXX
one
equal
isnt
maplong
attention
statement
checks
passed
index
arguments
bool
cachefinds
returning
storing
sci
list
else
10
9
findnews
Remember
likely
Also
31
public
keyed
Something
8
see
30
Cacheend
s
19
3
1
set
code
carry
ints
create
27
sized
already
called
know
29
calls
jend
cacheresizessize
comp
element
mapstring
simply
found
derived
vectorbool
sure
answer
declare
s0
increment
parameters
strings
search
unorderedmap
yes
write
rvgcnews
function
cachesize
beginning
17
pay
ji
dont
variable
22
quit
cachefindssecond
cacheresizes
26
implement
vecotr
cacheinsertmakepairs
6
uses
cachenz
nefore
od
given
memocounts