ESC101 - QUIZ 3 Session 2

Total points (2)



The quiz will be conducted in two sessions of 20 minutes each with a gap of 10 minutes in between two sessions.

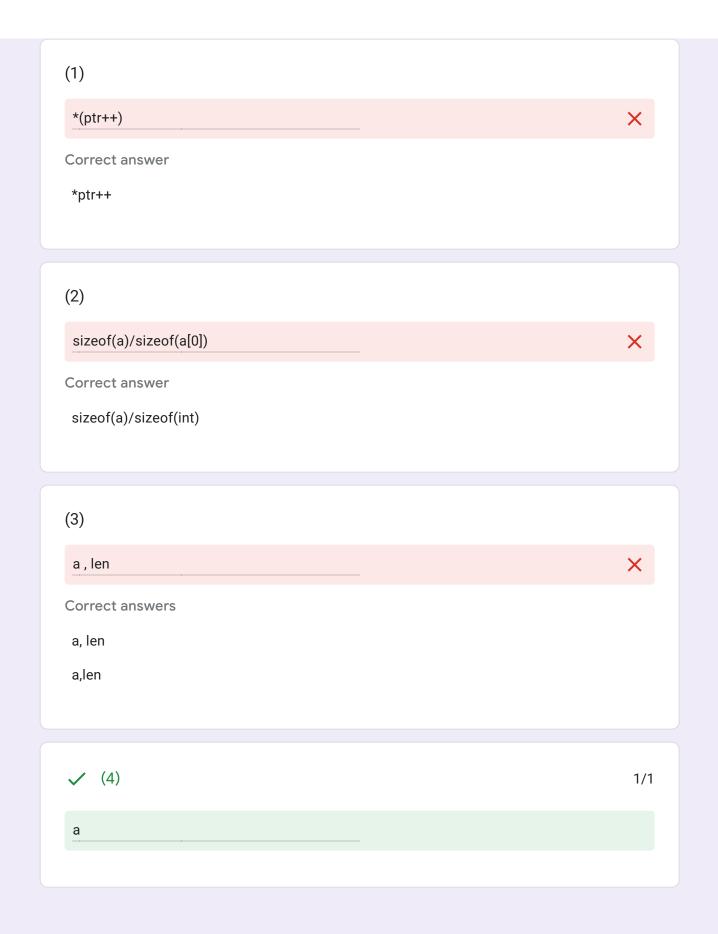
Read all instructions mentioned in the problem statement carefully before attempting it and try to keep your answers precise. Make sure to submit your response on time. Autosubmission is not available and if you fail to submit on time, you will get zero marks. If there are any issues like internet/power outage contact your Tutor ASAP.

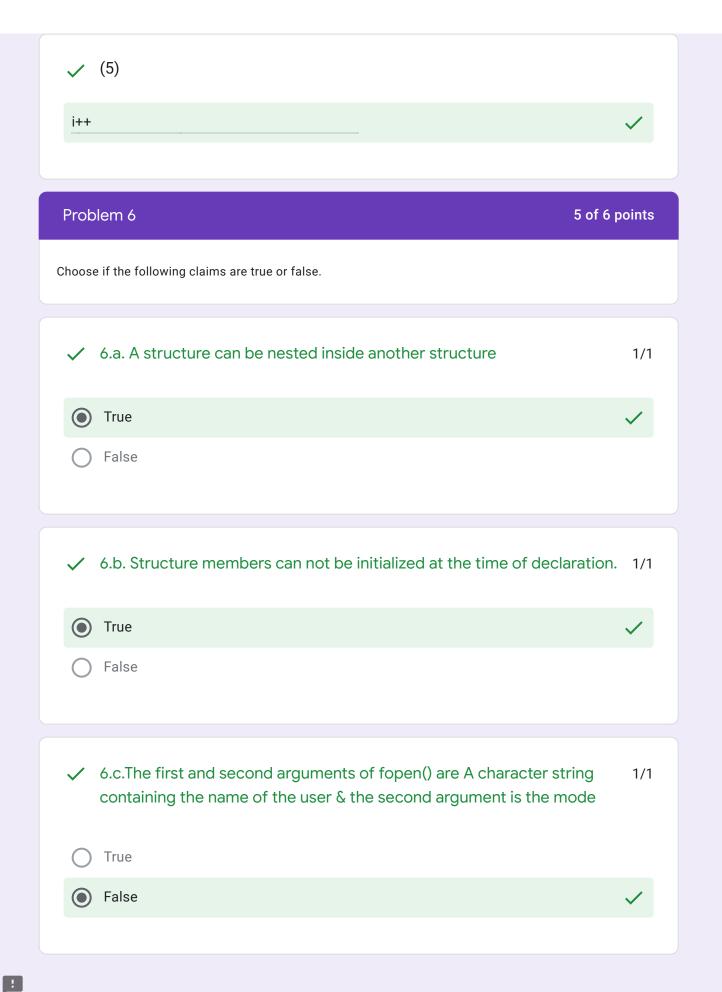
Problem 5

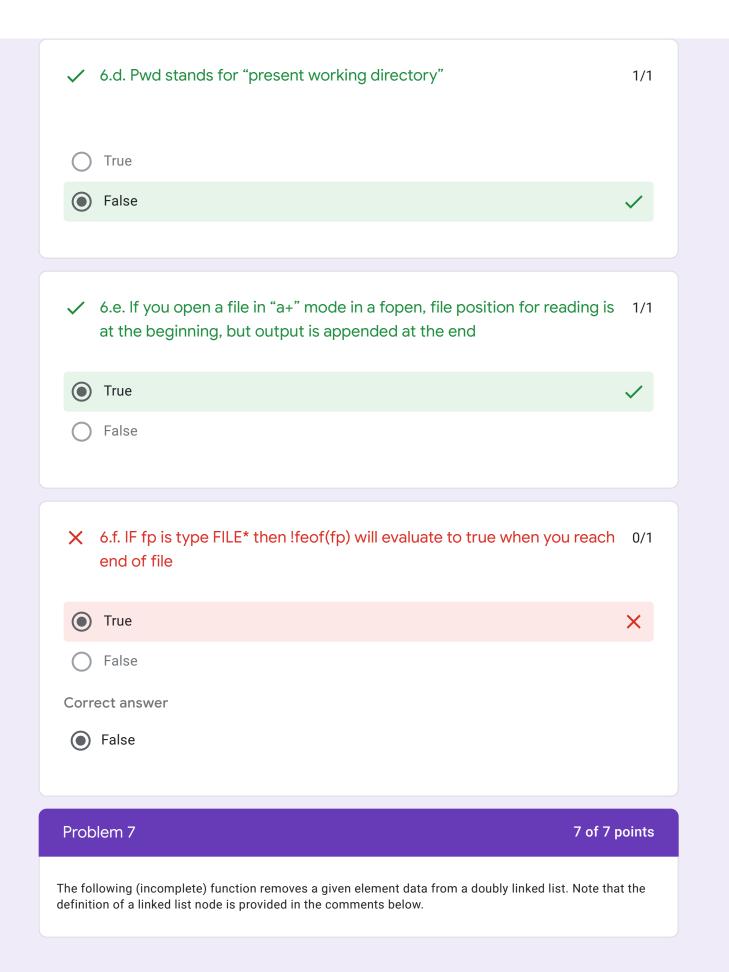
Complete the code

Fill in the blanks so that the output is 1 3 7 13 21... for array contents 1,4,9,16,25....sep

```
1 - void sequence(int * ptr, int d) {
     int x;
     for (x = 0; x < d; x++)
 3
       ____(1)___ = * ptr - x;
   }
5
 6
7 - main() {
     int a[];
 8
     /* Assume that array is initialized with
9 -
                 a[] = 1,4,9,16,25....*/
10
     int * k, i, len;
11
     /* len holds the number of elements in a[]*/
12
13
     len = ___(2)___;
     sequence(____(3)____);
14
     k = _{(4)};
15
     for (i = 0; i < len;)</pre>
16
       printf("%d", *(k + ____(5)___));
17
18 }
```







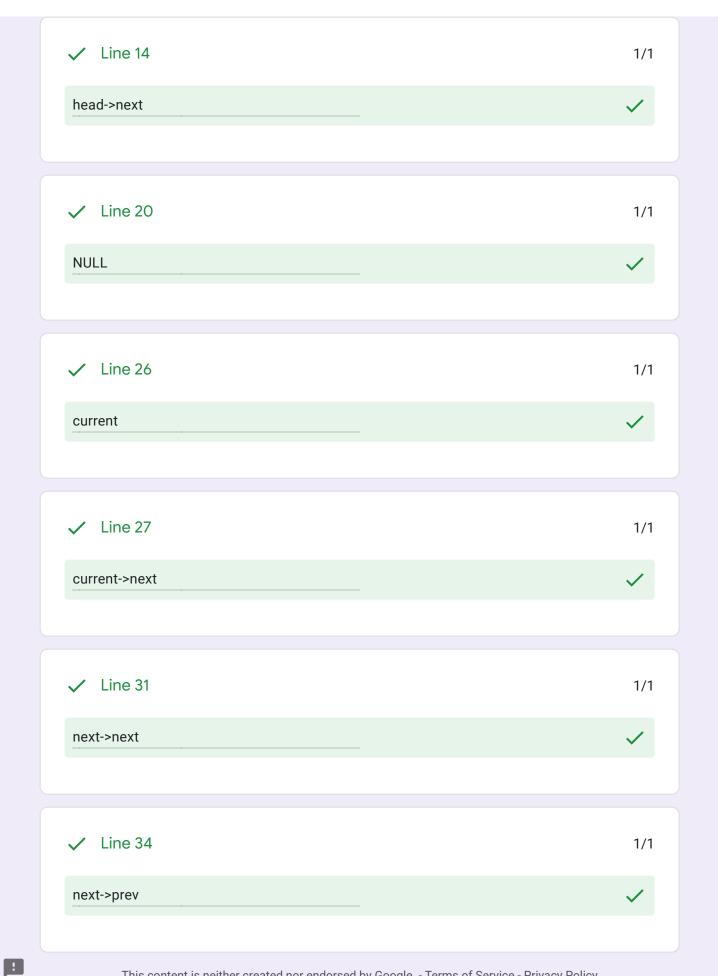
Complete the code so that it does the required task correctly.

```
1 /*
2 struct node {
int data; struct node *prev; struct node *next;
4 };
5 */
6
7 void remove_node(int data) {
   int pos = 0;
   struct node *pre_node;
9
if (head == NULL) return; // head stores pointer to the head of
        linked list
   if(head->data == data) {
11
       if(head->next != NULL) {
12
13
          head->____ = NULL;
          head = _____;
14
15
          return;
      } else {
16
          head = NULL;
17
18
         return;
       }
19
    } else if(head->data != data && head->next == ______
       return;
21
22
    struct node *current = head;
23
24
    while(current->next != NULL && current->data != data) {
26
       pre_node = _____
27
       current = ____
28
29
    if(current->data == data) {
       pre_node -> next = pre_node ->_____;
31
32
      if(pre_node->next != NULL) {
                                           // link back
33
          pre_node->____ = pre_node;
34
       }
       free(current);
36
37
    } else
       printf("%d not found in the list.", data);
38
39 }
```

✓ Line 13

next->prev





Google Forms