CS2050 – C Programming Quiz 5 SPRING 2017

There are 10 questions on this quiz. DO NOT PUT YOUR ANSWERS ON THIS SHEET – RECORD THEM ON THE ANSWER SHEET ONLY.

1. #include <stdio.h>
2. #include <stdlib.h>
3. void main() {
4. struct node
5. {
6. int num;
7. struct node \*ptr;
8. };
9. typedef struct node NODE;
11. NODE \*head, \*first = NULL, \*temp = NULL;
12. int count = 0;
13. int choice = 1;
14. while (choice) {
15. head = (NODE \*)malloc(sizeof(NODE));
16. printf("Enter the data item\n");
17. scanf("%d", &head-> num);
18. if (first != NULL) {
19. temp->ptr = head;
20. temp = head; }
21. else {
22. first = temp = head; }
23. printf("Do you want to continue?\n");
24. scanf("%d", &choice);
26. }
27. temp->ptr = 0;
28. temp = first;
29. printf("\n status of the linked list is\n");
30. while (temp != 0) {
31. printf("%d=>", temp->num);
32. count++;
33. temp = temp -> ptr;
34. }
35. printf("NULL\n");
36. printf("No. of nodes in the list = %d\n", count);
37. }  
      
    1. T / F This code will insert nodes in a sorted order.  
    2. What value will terminate the insertion of nodes into the list?  
    3. Assume an int = 8 bytes and a pointer = 8 bytes. How many bytes for a node?  
    4. List the name(s) of the “self referencing” pointer.  
    5. List the name(s) of the “utility” pointer.  
    6. How often does this code print out the list?   
     A) on every insertion B) at the end C) never  
    7. T / F This code handles a failed malloc() call.  
    8. \_\_\_\_\_\_\_\_\_\_\_\_ handle insertions more efficiently than \_\_\_\_\_\_\_\_\_\_\_\_ .  
      
      
    CS2050 – C Programming Quiz 5 NAME \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Spring 2017 LAB SECTION \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ANSWERS

1. T / F

2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7. T / F

8. Linked Lists / Arrays (circle one) handle insertions more efficiently than Linked Lists / Arrays (circle one).