Consider the linked list shown below. Assume that the nodes are in the usual info-link form. Use this list to answer Exercises 1 through 6. If necessary, declare additional variables. (Assume that list, current, temp, trail, and last are pointers of type nodeType.)



1. What is the output, if any, of each of the following C++ statements?

1. cout << current->info;
2. current = current->link;

cout << current->info;

1. cout << temp->link->link->info;
2. trail->link = NULL;

cout << trail->info;

1. cout << last->link->info;

2. What is the value of each of the following relational expressions?

1. current->link == temp
2. temp->link->link->info == 50
3. trail->link->link == 0
4. last->link == NULL
5. list == current

3. What are the effects, if any, of each of the following C++ statements?

1. trail->link = NULL;
2. delete last;
3. temp->link = trail;
4. list->info = 19;
5. current = current->link;
6. current->link = temp->link;

4. Write C++ statements to do the following:

1. Set the info of the second node to 52.
2. Make current point to the node with info 10.
3. Make trail point to the node before temp.
4. Make temp point to an empty list.
5. Set the value of the node before trail to 36.
6. Write a while loop to make current point to the node with info 10.

5. Mark each of the following statements as valid or invalid. If a statement is invalid, explain why.

1. current = list;
2. temp->link->link = NULL;
3. trail->link = 0;
4. \*temp = last;
5. list = 75;
6. temp->link->info = 75;
7. current->info = trail->link;
8. \*list = \*last;
9. current = last;
10. cout << trail->link->link->info;

6. If the following C++ code is valid, show the output. If it is invalid, explain why.

current = temp->link;

trail = list;

temp = list->link;

trail = temp;

temp->link = current->link;

current = trail->link;

cout << trail->info << " " << current->info << endl;

7. Show what is produced by the following C++ code. Assume the node is in the usual info-link form with the info of the type int. (list, trail, and current are pointers of type nodeType.)

list = new nodeType;

list->info = 28;

trail = new nodeType;

trail->info = 33;

trail->link = list;

list->link = NULL;

current = new nodeType;

current->info = 62;

trail->link = current;

current->link = list;

list = trail;

current = list->link;

trail = current->link;

cout << list->info << " " << current->info << " "

<< trail->info << endl;

8. Show what is produced by the following C++ code. Assume the node is in the usual info-link form with the info of the type int. (list, trail, and current are pointers of type nodeType.)

current = new nodeType;

current->info = 72;

current->link = NULL;

trail = current;

current = new nodeType;

current->info = 46;

current->link = trail;

list = current;

current = new nodeType;

current->info = 52;

list->link = current;

current->link = trail;

trail = current;

current = new nodeType;

current->info = 91;

current->link = trail->link;

trail->link = current;

current = list;

while (current!= NULL)

{

cout << current->info << “ “;

current=current->next;

}