```
NODE<int> n46(46, &n35, &n19);
NODE<int> n10(10, &n101, &n12);
NODE<int> root(55, &n46, &n10);

Queue<[1]NODE<int>> theq;
theq.enqueue(root);


cout << endl << endl;
while ([2]!theq.empty())       {
    NODE<int> next = theq.front();
    [3]theq.dequeue();
    cout << next.value << " ";
    if (next.left != nullptr)
        theq.enqueue([4]*(next.left));
    if (next.right != nullptr)
        [5]theq.enqueue(*next.right);
}
cout << endl << endl;
return 0;
}
```

Select the correct C++ expression to fill each numbered place.

Fill [1] with:

(a) List<int>
(b) List<NODE<int> *>
(c) List<NODE<int> >
(d) NODE<int>

Fill [2] with:

(a) theq.empty()
(b) !theq.empty()

Fill [3] with:

(a) theq.erase(next);
(b) theq.dequeue();
(c) theq.enqueue();
(d) theq.pop_back();