Student Number:

Name:

Question 1

1.0 pts

What is the computational complexity of adding an item to a Queue in the worst case in terms of Big O notation?

○ O(1)

○ O(log n)

○ O(n)

○ O(n log n)

Question 2

1.0 pts

Consider the following operations on a stack

```
push(10);
push(5);
pop();
push(7);
pop();
pop();
push(24);
push(3);
```

After the completion of all operations, what will size() operation result in? (Note: C++ stacks have a size() method).

Question 3

1.0 pts

Consider the following operations on a queue

```
enqueue(9);
dequeue();
enqueue(7);
enqueue(2);
dequeue();
enqueue(11);
enqueue(6);
```

After completing all of the operations, what value is at the front of the queue?

Question 4

1.0 pts

What is the computational complexity of adding an item to a Stack using a Linked List based implementation in the worst case in terms of Big O notation?

○ O(1)

○ O(log n)

○ O(n)

○ O(n^2)

Question 5

1.0 pts

Which of the following statements about linked lists and arrays are TRUE?

□ Both data structures store elements sequentially (contiguously) in memory

□ Both are linear data types

□ Linked Lists are more efficient if you need random access

□ Linked Lists are more efficient if you have to access elements with O(1) time

□ Both data structures can use iterators

Question 6

1.0 pts

What is the computational complexity of deleting an element, **e** from a doubly linked list with tail in the worst case in terms of Big O notation? Assume the list has n items.

○ O(n)

○ O(n^2)

○ O(1)

○ O(logn)

Question 7

1.0 pts

Which of the following container(s) is/are List ADT implementation(s) in C++? [Select all that apply]

□ Vector

□ Forward List

□ main()

□ Iterator

□ Integer

□ Array

Question 8

1.0 pts

Doubly linked lists allow random access in the container in constant time.

○ True

○ False

Question 9 1.0 pts

State the output of the following C++ program. If the program has a compiler or runtime error, write **error** in the box; if the program demonstrates undefined behavior, write **undefined** in the box).

```
#include <iostream>
#include <list>

int main()
{
    std::list<int> list_container {1, 2, 3, 4, 5};

    auto it = list_container.end();

    while(it != list_container.begin())
        std::cout << *(--it);

    return 0;
}
```

Question 10 1.0 pts

What type of iterator does the container in the following C++ code implement [Select the broadest category]:

```
#include <iostream>
#include <list>

int main()
{
    std::list<int> list_container {1, 2, 3, 4, 5};
    return 0;
}
```

○ Input/Output

○ Forward

○ Bidirectional

○ Random access