(2) (6%) Consider a deque of ints implemented by an int array of size MAXSIZE. In this implementation, an attribute front of type int points to the front of the deque. In the operations (or methods) that modify the deque, front needs to be updated. What types of changes to front should be supported? How should they be implemented in code?

1. Correctness: at most one error or none

class Foo {

int size;

public:

Foo(int s): size(s){ }

friend int getSize(Foo f);

// other unrelated methods omitted

};

int Foo::getSize(Foo f){

return f.size;

}

1. Correctness: at most one error or none

class A{

int a;

public:

A(int \_a=0): a(\_a){ }

int getA(){ return a; }

};

class B: public A{

int b;

public:

B(int \_b=0): b(\_b){ }

int getB() { return b; }

};

int main() {

A \*pa = new B();

B \*pb = dynamic\_cast<B \*>(pa);

return pb->getB();

}

Solution:

Increment front and wrap it to the beginning when the value after increment is MAXSIZE. In code:

front = (front+1)%MAXSIZE;

Decrement front and wrap it to the end when the value after decrement is -1. In code:

front = (front+MAXSIZE-1)%MAXSIZE;

(Note that it cannot be front = (front-1)%MAXSIZE;)

Function getSize is not a member method of Foo.

dynamic\_cast cannot succeed because there is no virtual method.