

2000 AP[®] COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

2. This question involves reasoning about the code from the Large Integer Case Study. A copy of the code is provided as part of this exam.

- (a) Write the new `BigInt` public member function `IsOdd`, as started below. `IsOdd` should return `true` if the `BigInt` is odd; otherwise, it should return `false`.

You may NOT assume that the `%` or `%=` operators have been defined for the `BigInt` class.

Complete function `IsOdd` below.

```
bool BigInt::IsOdd() const
// postcondition: returns true if this BigInt is odd;
//                otherwise, returns false
```

- (b) Write the free function `Power`, as started below. `Power` returns the value of `base` to the `exp` power, that is base^{exp} , where $\text{exp} \geq 0$. For example, the call `Power(3, 5)` returns 243, which is 3^5 .

You must use the following algorithm.

```
Initialize a variable, product, to be 1.
While exp is not zero do the following:
    if exp is odd, product is set to product times the base
    square the base
    divide exp by two
When done, product contains the result.
```

Assume that a new member function, `DivBy2`, has been defined for the `BigInt` class, as specified below. `DivBy2` divides this `BigInt` by 2 (using integer division). (You do not need to write the body of `DivBy2`.)

```
void BigInt::DivBy2(); // this BigInt is divided by 2
```

In writing `Power`, you may use the `BigInt` public member function `DivBy2` specified above and you may use the `BigInt` public member function `IsOdd` specified in part (a). Assume that `IsOdd` works as specified, regardless of what you wrote in part (a).

Complete function `Power` below.

```
BigInt Power(const BigInt & base, const BigInt & exp)
// precondition: base > 0 and exp ≥ 0
// postcondition: returns the value of base to the exp
```