Total: 97/100

David Pick
CM 2403

I spent 3 hours on this assignment.

A)

```
  //@ assert true;
//@ assert 0 == (\sum int k; b.length - 1+1 <= k && k < b.length; b[k]) && -1 <= b.length - 1 &&
b.length - 1 < b.length;
i' = b.length - 1;
//@ assert 0 == (\sum int k; i'+1 <= k && k < b.length; b[k]) && -1 <= i' && i' < b.length;
s' = 0;
  //@ maintaining -1 <= i && i < b.length;
  //@ maintaining s == (\sum int k; i+1 <= k && k < b.length; b[k]);
  //@ decreasing i;

  //@ assert s' == (\sum int k; i+1 <= k && k < b.length; b[k]) && -1 <= i && i < b.length;
while (i != -1) {
  //@ assert s == (\sum int k; i+1 <= k && k < b.length; b[k]) && -1 <= i && i < b.length && i != -1;
  //@ assert s + b[i] == (\sum int k; i <= k && k < b.length; b[k]) + b[i] && -1 <= i - 1 && i - 1 <
b.length && i - 1 != -1
```

(-3) At this point, (i - 1) == -1 is a possibility. In fact, it's the only way to get out of the loop.

```
  s' = s + b[i];
  //@ assert s' ==(\sum int k; (i + 1) - 1 <= k && k < b.length; b[k]) && -1 <= i - 1 && i - 1 <
b.length && i - 1 != -1
  i' = i - 1;
  //@ assert s ==(\sum int k; i'+1 <= k && k < b.length; b[k]) && -1 <= i' && i' < b.length && i' != -1
}
  //@ assert s == (\sum int k; i+1 <= k && k < b.length; b[k]) && -1 <= i && i < b.length && i != -1
  //@ assert s ==(\sum int k; 0 <= k && k < b.length; b[k]) && -1 <= -1 && -1 < b.length && -1 ==
-1
  //@ assert s ==(\sum int k; 0 <= k && k < b.length; b[k]) && -1 < b.length
  //Arrays must have a length >= to 0
  //@ assert s ==(\sum int k; 0 <= k && k < b.length; b[k])
  //@ assert s == (\sum int k; 0 <= k && k < b.length; b[k]);
```

B)

Good

```
  //@ assert true;
  //@ assert (0 <= 0 && 0 <= b.length) && !(\exists int k; 0 <= k && k < 0; x == b[k]);
i' = 0;
  //@ maintaining 0 <= i && i <= b.length;
  //@ maintaining !(\exists int k; 0 <= k && k < i; x == b[k]);
  //@ decreasing -i;
  //@ assert (0 <= i' && i' <= b.length) && !(\exists int k; 0 <= k && k < i'; x == b[k]);
while ((i < b.length) && (x != b[i])) {
  //@ assert (0 <= i && i <= b.length) && !(\exists int k; 0 <= k && k < i; x == b[k]) && (i <
```

```
b.length) && (x != b[i]);
  // i + 1 <= b.length ==> i < b.length
  //@ assert (0 <= i + 1 && i + 1 <= b.length) && !(\exists int k; 0 <= k && k < i + 1; x == b[k]);
  i' = i + 1;
  //@ assert (0 <= i' && i' <= b.length) && !(\exists int k; 0 <= k && k < i'; x == b[k]);
}
  //@ assert (0 <= i && i <= b.length) && !(\exists int k; 0 <= k && k < i; x == b[k]) && !((i <
b.length) && (x != b[i]));
  //@ assert (0 <= i && i <= b.length) && !(\exists int k; 0 <= k && k < i; x == b[k]) && ((x == b[i]) ||
(i >= b.length));
  //@ assert (0 <= i && i < b.length && x == b[i]) || (i == b.length && !(\exists int k; 0 <= k && k <
b.length; x == b[k]));
  /*@ assert (0 <= i && i < b.length && x == b[i]) ||
    @       (i == b.length && !(\exists int k; 0 <= k && k < b.length; x == b[k]));
    @*/
```

C) Good

```
  //@ assert 0 < b.length;
  //1 <= b.length ==> 0 < b.length
  //j = 0 ==> b[0] == b[j]
  //@ assert (0 < 1 && 1 <= b.length) && (\forall int j; 0 <= j && j < 1; b[0] >= b[j]);
i' = 1;
  //@ assert (0 < i' && i' <= b.length) && (\forall int j; 0 <= j && j < i'; b[0] >= b[j]);
k' = 0;
  //@ maintaining 0 < i && i <= b.length;
  //@ maintaining (\forall int j; 0 <= j && j < i; b[k] >= b[j]);
  //@ decreasing -i;

  //@ assert (0 < i && i <= b.length) && (\forall int j; 0 <= j && j < i; b[k'] >= b[j]);
while (i < b.length) {
  //@ assert (0 < i && i <= b.length) && (\forall int j; 0 <= j && j < i; b[k] >= b[j]) && (i < b.length)
  // i + 1 <= b.length ===> i < b.length
  /*@ assert (0 < i + 1 && i + 1 <= b.length) && (\forall int j; 0 <= j && j < i + 1; b[i] >= b[j]) && (b[i]
>= b[k]) ||
    @         (0 < i + 1 && i + 1 <= b.length) && (\forall int j; 0 <= j && j < i + 1; b[i] >= b[j]) && (b[i]
< b[k])
    @*/
  if (b[i] >= b[k]) {
    //@ assert (0 < i + 1 && i + 1 <= b.length) && (\forall int j; 0 <= j && j < i + 1; b[i] >= b[j]) &&
(b[i] >= b[k]);
    k' = i;
    //@ assert (0 < i + 1 && i + 1 <= b.length) && (\forall int j; 0 <= j && j < i + 1; b[k'] >= b[j]);
  }
  //@ assert (0 < i + 1 && i + 1 <= b.length) && (\forall int j; 0 <= j && j < i + 1; b[k] >= b[j]);
  i' = i + 1;
```

```
  //@ assert (0 < i' && i' <= b.length) && (\forall int j; 0 <= j && j < i'; b[k] >= b[j]);
}
  //@ assert (0 < i && i <= b.length) && (\forall int j; 0 <= j && j < i; b[k] >= b[j]) && !(i < b.length)
  //@ assert (i == b.length) && (\forall int j; 0 <= j && j < i; b[k] >= b[j])
  //@ assert (\forall int j; 0 <= j && j < b.length; b[k] >= b[j]);
```