ESC/Java “demo”

```
class Bag {
  int[] a;
  int   n;
  int extractMin() {
   int m = Integer.MAX_VALUE;
   int mindex = 0;
   for (int i = 1; i <= n; i++) {
        if (a[i] < m) { mindex =i; m = a[i]; } }
   n--;
   a[mindex] = a[n];
   return m;
 }
```

ESC/Java “demo”

```
class Bag {
  int[] a;
  int   n;
  int extractMin() {
   int m = Integer.MAX_VALUE;
   int mindex = 0;
   for (int i = 1; i <= n; i++) {
        if (a[i] < m) { mindex =i; m = a[i]; } }
   n--;
   a[mindex] = a[n];
   return m;
 }
```

Warning: possible null deference. Plus other warnings

ESC/Java “demo”

```
class Bag {
  int[] a;  //@ invariant a != null;
  int   n;
  int extractMin() {
   int m = Integer.MAX_VALUE;
   int mindex = 0;
   for (int i = 1; i <= n; i++) {
        if (a[i] < m) { mindex =i; m = a[i]; } }
   n--;
   a[mindex] = a[n];
   return m;
 }
```

ESC/Java “demo”

```
class Bag {
  int[] a;  //@ invariant a != null;
  int   n;
  int extractMin() {
   int m = Integer.MAX_VALUE;
   int mindex = 0;
   for (int i = 1; i <= n; i++) {
        if (a[i] < m) { mindex =i; m = a[i]; } }
   n--;
   a[mindex] = a[n];
   return m;
 }
```

Warning: Array index possibly too large

ESC/Java “demo”

```
class Bag {
  int[] a;  //@ invariant a != null;
  int   n;  //@ invariant 0 <= n && n <= a.length;
  int extractMin() {
   int m = Integer.MAX_VALUE;
   int mindex = 0;
   for (int i = 1; i <= n; i++) {
        if (a[i] < m) { mindex =i; m = a[i]; } }
   n--;
   a[mindex] = a[n];
   return m;
 }
```

ESC/Java “demo”

```
class Bag {
  int[] a;  //@ invariant a != null;
  int   n;  //@ invariant 0 <= n && n <= a.length;
  int extractMin() {
   int m = Integer.MAX_VALUE;
   int mindex = 0;
   for (int i = 1; i <= n; i++) {
        if (a[i] < m) { mindex =i; m = a[i]; } }
   n--;
   a[mindex] = a[n];
   return m;
 }
```

Warning: Array index possibly too large

ESC/Java “demo”

```
class Bag {
  int[] a;  //@ invariant a != null;
  int   n;  //@ invariant 0 <= n && n <= a.length;
  int extractMin() {
   int m = Integer.MAX_VALUE;
   int mindex = 0;
   for (int i = 0; i < n; i++) {
        if (a[i] < m) { mindex =i; m = a[i]; } }
   n--;
   a[mindex] = a[n];
   return m;
 }
```

ESC/Java “demo”

```
class Bag {
  int[] a;  //@ invariant a != null;
  int   n;  //@ invariant 0 <= n && n <= a.length;
  int extractMin() {
   int m = Integer.MAX_VALUE;
   int mindex = 0;
   for (int i = 0; i < n; i++) {
        if (a[i] < m) { mindex =i; m = a[i]; } }
   n--;
   a[mindex] = a[n];
   return m;
 }
```

Warning: Possible negative array index

ESC/Java “demo”

```
class Bag {
  int[] a;  //@ invariant a != null;
  int   n;  //@ invariant 0 <= n && n <= a.length;
  //@ requires n > 0;
  int extractMin() {
   int m = Integer.MAX_VALUE;
   int mindex = 0;
   for (int i = 0; i < n; i++) {
        if (a[i] < m) { mindex =i; m = a[i]; } }
   n--;
   a[mindex] = a[n];
   return m;
 }
```

ESC/Java “demo”

```
class Bag {
  int[] a;  //@ invariant a != null;
  int   n;  //@ invariant 0 <= n && n <= a.length;
  //@ requires n > 0;
  int extractMin() {
   int m = Integer.MAX_VALUE;
   int mindex = 0;
   for (int i = 0; i < n; i++) {
        if (a[i] < m) { mindex =i; m = a[i]; } }
   n--;
   a[mindex] = a[n];
   return m;
  }
```

No more warnings about this code