TEMILOLUWA PHILIP OJO A8 TASK 2.

Method

```
public static int max(final int... array) {

   // Validates input
    if(array == null || array.length == 0)
      throw new IllegalArgumentException();
    // Finds and returns max
    int max = array[0];
    for (int j = 1; j < array.length; j++) {
      if (max != Integer.MAX_VALUE && array[j] >= max + 1) {
        max = array[j];
      }
    }
    return max;
  }
```

Equivalent mutant

- **Operator used**
  A++ → ++A

- **Line modified**

```
for (int j = 1; j < array.length; j++)
```

- **Line obtained**

```
for (int j = 1; j < array.length; ++j)
```

- **Justification**

A mutant is said to be equivalent to the original program if the mutant generated doesn't get killed by the tests.

Not Valid

- Operator used:
  == → **=**

- **Line modified**

```
if(array == null || array.length == 0)
```

- **Line obtained**

```
if(array = null || array.length = 0)
```

Mutant generated: **if(array = null || array.length = 0) …**

- **Justification**

A mutant is not valid if the mutation operand applied to it prevents the program from compiling.

Changing **==** to **=** in the method below will prevent the code from compiling thereby producing a not valid mutant.

Useful

- **Operator used**
  != → **==** (negate conditionals)

- **Line modified**

```
    if (max != Integer.MAX_VALUE && array[j] >= max + 1) {
```

- **Line obtained**

```
  if (max == Integer.MAX_VALUE && array[j] >= max + 1) {
```

- **Justification**

Typically a mutant is useful if the mutant is killed by less or equal to 2 test cases. Negating the **max != Integer.MAX_VALUE** condition will result in two tests failing or not being killed b

Not Useful

- **Operator used**
  >= → <

- **Line Modified**

```
if (max != Integer.MAX_VALUE && array[j] >= max + 1
```

- **Line Obtained**

```
if (max != Integer.MAX_VALUE && array[j] < max + 1
```

- **Justification**

A mutant is not useful if it's killed by greater or equal to 3 test cases. Applying negate conditional mutator operand to the line statement above produces 3 mutants that are killed by the operator applied.