Testing Your Interpreter, Part 2

Here are some tests for your interpreter.

Test 1: This code should return 20.

```
var x = 10;
{
  var y = 2;
  var z = x * y;
  x = z;
}
return x;
```

Test 2: This code should return 164.

```
var a = 31160;
var b = 1476;
if (a < b) {
  var temp = a;
  a = b;
  b = temp;
}
var r = a % b;
while (r != 0) {
  a = b;
  b = r;
  r = a % b;
}
return b;
```

Test 3: This code should return 32.

```
var x = 0;
var y = 10;
while (!(x >= y) || !(y > 25)) {
  x = x + 2;
  y = y + 1;
}
return x;
```

Test 4: This code should return 2.

```
var x = 1;
var y = x + 1;
if (x < y) {
  var z = 10;

  if (x < z) {
    var swap = y;
    y = x;
    x = swap;
  }
}
return x;
```

Test 5: This code should give an error.

```
var x = 10;
var y = 4;
if (x < y) {
  var min = x;
}
```

```
else {
   var min = y;
}
return min;
```

Test 6: This code should return 25.

```
var x = 0;
x = x + 25;
return x;
x = x + 25;
return x;
x = x + 25;
return x;
```

Test 7: This code should return 21.

```
var x = 0;
var result = 0;

while (x < 10) {
   if (result > 15) {
      return result;
   }
   result = result + x;
   x = x + 1;
}
return result;
```

Test 8: This code should return 6.

```
var x = 0;
while (x < 6) {
   x = x + 1;
   continue;
   x = x + 100;
}
return x;
```

Test 9: This code should return -1.

```
var x = 0;
while (x < 10) {
   x = x - 1;
   break;
   x = x + 100;
}
return x;
```

Test 10: This code should return 789.

```
var x = 0;
var y = x;
var z = y;
while (1 == 1) {
   y = y - x;
   while (2 == 2) {
      z = z - y;
      while (3 == 3) {
         z = z + 1;
         if (z > 8)
            break;
         else
            continue;
```

```
    }
    y = y + 1;
    if (y <= 7)
      continue;
    else
      break;
  }
  x = x + 1;
  if (x > 6)
    break;
  else
    continue;
}
return x * 100 + y * 10 + z;
```

Test 11: This code should give an error.

```
var x = 0;
while (x < 10) {
  var y = 0;
  x = x + 1;
  y = y - 1;
  break;
}
if (x > 0) {
  x = y;
}
return x;
```

Test 12: This code should give an error.

```
var x = 1;
var y = 2;
if (x < y) {
  var z = 0;
  while (z < 100) {
    var a = 1;
    z = z + a;
    continue;
    z = 1000;
  }
  if (z != x) {
    z = a;
  }
}
return x;
```

Test 13: This code should give an error.

```
var x = 1;
break;
return x;
```

Test 14: This code should return 12.

```
var x = 1;
while (true) {
  x = x + 1;
  if (x > 10 && x % 2 == 0)
   break;
}
return x;
```

Test 15 should return 125.

```
var x;

try {
  x = 20;
  if (x < 0)
    throw 10;
  x = x + 5;
}
catch(e) {
  x = e;
}
finally {
  x = x + 100;
}
return x;
```

Test 16 should return 110.

```
var x;

try {
  x = 20;
  if (x > 10)
    throw 10;
  x = x + 5;
}
catch(e) {
  x = e;
}
finally {
  x = x + 100;
}
return x;
```

Test 17 should return 2000400.

```
var x = 0;
var j = 1;

try {
  while (j >= 0) {
    var i = 10;
    while (i >= 0) {
      try {
        if (i == 0)
          throw 1000000;
        x = x + 10*i / i;
      }
      catch(e) {
        if (j == 0)
          throw 1000000;
        x = x + e / j;
      }
      i = i - 1;
    }
    j = j - 1;
  }
}
catch (e2) {
  x = x * 2;
}
return x;
```

Test 18 should return 101.

```
var x = 10;
```

```
var result = 1;

try {
  while (x < 10000) {
    result = result - 1;
    x = x + 10;

    if (x > 1000) {
      throw x;
    }
    else if (x > 100) {
      break;
    }
  }
}
finally {
  result = result + x;
}
return result;
```

Test 19 should give an error.

```
var x = 10;
var result = 1;

try {
  while (x < 10000) {
    result = result - 1;
    x = x * 10;

    if (x > 1000)
      throw x;
  }
}
catch (ex) {
  throw 1;
}
return result;
```

Additional Test for Those Seeking an Extra Challenge

Test 20: This code should return 21.

```
var x = 0;
while ((x = x + 1) < 21)
  x = x;
return x;
```