Lab 12 Parameters and variables & Shell functions

1. Rewrite function isvalidip using parameter expansion instead of changing IFS. Then write another script that accepts a list of IP address (the number of IP address is unknown) and calls the function you have just wrote to check for the validity of IP address(es) and finally print a list of valid IP address followed by a list of invalid IP address.

Example 1

```
$ ./checkIPvalidity 127.0.0.1 168.260.0.234 1.2.3.4 123.1oo.34.21 204.225.122.150
Valid IP address are as follows.
127.0.0.1
1.2.3.4
204.225.122.150
Invalid IP address are as follows.
168.260.0.234
123.1oo.34.21
```

Example 2

```
$ ./checkIPvalidity 161.200.126.68 161.200.122.244 161.200.107.79
Valid IP address are as follows.
161.200.126.68
161.200.122.244
161.200.107.79
```

Example 3

```
$ ./checkIPvalidity 161.200.126.368 161.200.122.344 161.300.107.79
Invalid IP address are as follows.
161.200.126.368
161.200.122.344
161.300.107.79
```

Source: Modified from Johnson, C. and Varma, J. (2015) Pro bash programming. Apress.

2. Add a check to max3 to verify that VARNAME is a valid name for a variable.

```
$ max3 32 44 13 results
$ echo ${results[@]}
44 32 13
$ max3 32 44 13 re+sults
Invalid variable name
$ max3 32 44 13 res^ults
Invalid variable name
$ max3 32 44 13 8results
Invalid variable name
```

Source: Johnson, C. and Varma, J. (2015) Pro bash programming. Apress.