```
#!/usr/bin/env python3

import unittest
from emails import find_email


class EmailsTest(unittest.TestCase):
  def test_basic(self):
    testcase = [None, "Bree", "Campbell"]
    expected = "breee@abc.edu"
    self.assertEqual(find_email(testcase), expected)

  def test_one_name(self):
    testcase = [None, "John"]
    expected = "Missing parameters"
    self.assertEqual(find_email(testcase), expected)

  def test_two_name(self):
    testcase = [None, "Roy","Cooper"]
    expected = "No email address found"
    self.assertEqual(find_email(testcase), expected)

if __name__ == '__main__':
  unittest.main()
```

```
#!/usr/bin/env python3

import csv
import sys


def populate_dictionary(filename):
  """Populate a dictionary with name/email pairs for easy lookup."""
  email_dict = {}
  with open(filename) as csvfile:
    lines = csv.reader(csvfile, delimiter = ',')
    for row in lines:
      name = str(row[0].lower())
      email_dict[name] = row[1]
  return email_dict

def find_email(argv):
  """ Return an email address based on the username given."""
  # Create the username based on the command line input.
  try:
    fullname = str(argv[1] + " " + argv[2])
    # Preprocess the data
    email_dict = populate_dictionary('/home/{{ username
}}/data/user_emails.csv')
```

```
    # If email exists, print it
    if email_dict.get(fullname.lower()):
      return email_dict.get(fullname.lower())
    else:
      return "No email address found"
  except IndexError:
    return "Missing parameters"

def main():
  print(find_email(sys.argv))

if __name__ == "__main__":
  main()
```