Main Program with Command Line Arguments

```python
import sys

def add(a, b):
    return a+b

if __name__=='__main__':
    first = int(sys.argv[1])
    second = int(sys.argv[2])
    sum = add(first, second)
    print(str(sum))
```

stored in try.py

python try.py 4 5

9
Exercise:
Write a Python program to read positive integers one at a time from the user, ending when he/she types -1.

Put those between 1 and 100 into a list called mylist.

Then print mylist.

How to read:
num = int(input('Enter integer (-1 to quit): '))
Exercise: Program to read positive integers, and put those between 1 and 100 into a list

```python
#
terminate = False
mylist = []
# prompt for integers
while not terminate:
    #
    num = int(input('Enter integer (-1 to quit): '))
    if num == -1:
        terminate = True
    else:
        if num >= 1 and num <= 100:
            mylist.append(num)
    #
print('My List')
for i in mylist:
    print(str(i))
```
Global Test 1

```python
>>> x = 5
>>> y = 6
>>> z = 7
>>> def foo(x):
...    global y
...    z = x + y
...    y = y + 1
...    return z
...
>>> q = foo(2)
>>> q
8
>>> x
5
>>> y
7
>>> z
7
```

- `y` has changed, since it was declared `global` in the function.
- `z` has not changed, since the `z` in the function is `local`. 
Global Test 2

```python
>>> count = 0
>>> def f1():
...    global count
...    count = count + 1
...
>>> def f2():
...    global count
...    count = count + 2
...
>>> count
0
>>> f1()
>>> count
1
>>> f2()
>>> count
3
>>> f1()
>>> count
4
```