Name:

YOU MAY READ THIS PAGE BEFORE THE EXAM BEGINS

1. You have 75 minutes to create and submit the 3 programs in this exam.
2. When instructed to start, download and extract the following PyCharm Project under the `~/workspace` folder if you do not have it there already.

   `https://www.cs.umb.edu/~siyer/teaching/ipp.zip`

3. Open the Project in PyCharm. To create a program, right-click on `ipp` in the top-left window and then select the *New → Python File* menu. Enter the name of the program in the pop-up window. Note that the name is case-sensitive and must match the suggested name exactly.
4. You may use the text, your notes, your code from the assignments, and the code on the CS110 course website. No form of communication is permitted (eg, talking, texting, etc.) during the exam, except with the course staff.
5. Submit your programs (`.py` files) on Gradescope under the assignment named `Sample Programming Exam 1`.
6. Return this exam sheet to the course staff with your name written at the top. Failing to do so will void your exam submission on Gradescope.
7. You are *not* allowed to leave the exam hall before the official end time even if you are done early.
8. Your programs will be graded based on correctness, clarity, and efficiency.
9. Discussing the exam contents with anyone who has not taken the exam is a violation of the academic honesty code.

DO NOT READ FURTHER UNTIL SO INSTRUCTED

Problem 1. (5 Points) Write a program called `cylinder.py` that accepts r (float) and h (float) as command-line arguments, and writes to standard output the surface area a and volume v of a cylinder having base radius r and height h, which are computed as $a = 2\pi r(r + h)$ and $v = \pi r^2 h$.

cylinder.py

```
import math
import stdio
import sys

...
```

~/workspace/ipp

```
$ python3 cylinder.py 1 1
area = 12.566370614359172
volume = 3.141592653589793
```

Problem 2. (10 Points) Write a program called `sum.py` that accepts integers from standard input and writes the sum of squares of the *even* integers to standard output. For example, if the inputs are 5, 6, 7, and 8, the program writes $6^2 + 8^2 = 100$.

sum.py

```
import stdio

...
```

~/workspace/ipp

```
$ python3 sum.py
5 6 7 8
<ctrl-d>
100
```

Relevant standard input functions: `stdio.isEmpty()`, `stdio.readInt()`.

Problem 3. (10 Points) Implement a library called `listops.py` that supports the following API:

listops

`any(a)`	returns `True` if any value in the list `a` is `True`, and `False` otherwise
`all(a)`	returns `True` if all values in the list `a` are `True`, and `False` otherwise
`exactly(a, k)`	returns `True` if exactly `k` values in the list `a` are `True`, and `False` otherwise
`count(a)`	returns the number of `True` values in the list `a`

listops.py

```
def any(a):
    ...

def all(a):
    ...

def exactly(a, k):
    ...

def count(a):
    ...

# Unit tests the library.
def _main():
    import stdio

    a = [False, False, True, False, True, True, True]
    stdio.writeln(any(a))
    stdio.writeln(all(a))
    stdio.writeln(exactly(a, 3))
    stdio.writeln(count(a))

if __name__ == "__main__":
```

```
    _main()
```

```
>_ ~/workspace/ipp
$ python3 listops.py
True
False
False
4
```

Files to Submit

1. `cylinder.py`
2. `sum.py`
3. `listops.py`

Solutions

cylinder.py

```python
import math
import stdio
import sys

# Accept r (float) and h (float) as command-line arguments.
r = float(sys.argv[1])
h = float(sys.argv[2])

# Calculate surface area (a) and volume (v) of cylinder with base radius r and height h.
a = 2 * math.pi * r * (r + h)
v = math.pi * r ** 2 * h

# Write a and v to standard output.
stdio.writeln("area = " + str(a))
stdio.writeln("volume = " + str(v))
```

sum.py

```python
import stdio

# Set total to 0.
total = 0

while not stdio.isEmpty():
    # Until standard input is not empty ...

    # Read an integer x.
    x = stdio.readInt()

    # Add x * x to total if x is even.
    total += x * x if x % 2 == 0 else 0

# Write total to standard output.
stdio.writeln(total)
```

listops.py

```python
# Returns True if any value in the list a is True, and False otherwise.
def any(a):
    return count(a) > 0


# Returns True if all values in the list a are True, and False otherwise.
def all(a):
    return count(a) == len(a)


# Returns True if exactly k values in the list a are True, and False otherwise.
def exactly(a, k):
    return count(a) == k


# Returns the number of True values in the list a.
def count(a):
    total = 0
    for v in a:
        if v:
            total += 1
    return total


# Unit tests the library. [DO NOT EDIT]
def _main():
    import stdio

    a = [False, False, True, False, True, True, True]
    stdio.writeln(any(a))
    stdio.writeln(all(a))
    stdio.writeln(exactly(a, 3))
    stdio.writeln(count(a))


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