

```
""" superhero.py
demonstrates using a for() structure
to step through a list
3/22/06 """
```

```
heroes = [  
    "Buffalo Man",  
    "Geek Boy",  
    "Wiffle-Ball Woman"  
]
```

```
for hero in heroes:  
    print "Never fear,", hero, "is here."
```

```
""" minivan.py
    Simulates a car trip with small children
    """

tripFinished = "NO"
while tripFinished != "YES":
    tripFinished = raw_input("Are we there yet? ")
    tripFinished = tripFinished.upper()

print "Can we go home now?"
```

```
""" debug.py
Demonstrates pdb command-line debugger
```

THIS PROGRAM DOES NOT RUN NORMALLY!

```
import the pdb library
place pdb.set_trace() function where you want debugging
to start
```

Interactive debugging commands:

```
h - help
l - list the current code fragment
n - next command (skips functions)
s - step to next command (goes into functions)
print x - print the current value of the variable x
c - continue execution
!statement - executes the statement as python
q - quit debugger
```

Don't forget to remove set_trace before releasing program

```
"""
```

```
import pdb;
```

```
def main():
```

```
    pdb.set_trace()
```

```
    j = 3
```

```
    for i in range(10):
```

```
        j = j + 10
```

```
        if j > 40:
```

```
            j = 1
```

```
if __name__ == "__main__":
```

```
    main()
```