

Practice Set #1 Solutions

1) Create a list of numbers from 5 to 15 inclusive stepping by 0.5

Optimal solution:

```
numbers=[i/2.0 for i in xrange(10,31)]
```

Other solutions:

```
numbers=[]
```

```
for i in xrange(21):
```

```
    numbers.append(i/2.0+5.0)
```

```
numbers=[]
```

```
n=5.0
```

```
while n<=15.0:
```

```
    numbers.append(n)
```

```
    n+=0.5
```

2) Start with the string "this is a short test string" and create a new string with the letters sorted alphabetically

optimal:

```
print "".join(sorted("this is a test string"))    # sorted() implicitly converts to a list
```

Other solutions:

```
s="this is a test string"
```

```
l=list(s)
```

```
l.sort()
```

```
print "".join(l)
```

3) Create a string containing only the unique letters in "abracadabra"

Optimal solution:

```
print "".join(set("abracadabra"))
```

Other solutions:

```
# quite inefficient
```

```
s="abracadabra"
```

```
unique=[]
```

```
for letter in s:
```

```
    if not letter in unique : unique.append(letter)
```

```
print "".join(unique)
```

```
# even more inefficient since a new unique string is created with each new letter
```

```
s="abracadabra"
```

```
unique=""
```

```
for letter in s:
```

```
    if not letter in unique : unique+=letter
```

```
print unique
```

```
# This one at least uses a set, which is more efficient
s="abracadabra"
unique=set()
for letter in s: unique.add(letter)
print "".join(unique)
```

4) Start with `s="1 2 4 8 16"`. Convert the string to a list of integers and take the log base 2 of each number.

Optimal solution:
`newlist=[log(float(i),2) for i in "1 2 4 8 16".split()]`

Other solutions:

```
s="1 2 4 8 16"
l=s.split()
for i in xrange(len(l)): l[i]=log(float(l[i]),2)
```