Exposure CS 2022 for CS1

Chapter 17 Slides

Sequential Text Files

PowerPoint Presentation created by:
Mr. John L. M. Schram
and Mr. Leon Schram
Authors of Exposure
Computer Science

Section 17.1

Introduction

Programs and Data Files

Data information processed by word processing, spreadsheet and similar application programs are stored as data files.

Python programs, written with a text editor or some Integrated Development Environment (*IDE*), are also data files.

Programs have the capability to create and retrieve their own data files. This means that data entered in any one of your programs can be saved for later use.

Data Organization

bit	This is the fundamental building block of all computer information – a **bi**nary digi**t** – which stores only a **1** or **0**.
byte	One *byte* equals 8 *bits*. The *ASCII* format uses 1 byte to store a character. *UTF-8* uses between 1 and 4 bytes to store a character.
field	A *field* is one specific unit of information. Examples: **name**, **address**, **gpa**, **salary**, **diagnosis**
record	A *record* consists of a set of *fields* for a specific purpose. Examples of the fields contained by different records: **student** record: **name**, **address**, **gpa**, **classRank** **employee** record: **name**, **address**, **salary**, **position** **patient** record: **name**, **address**, **diagnosis**, **allergies**
file	A data base *file* consists of a sequence of records. Example: A school can have a *file* of **student** records.

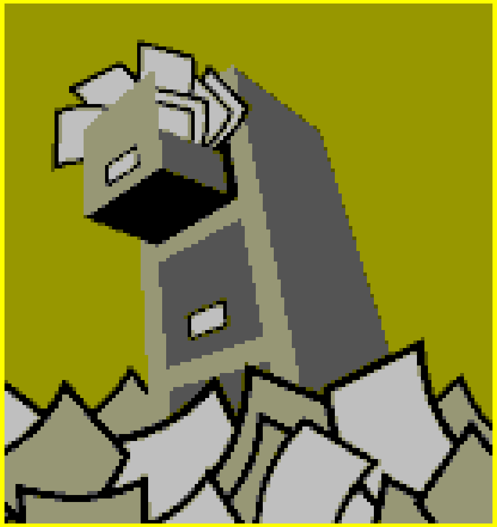
Section 17.2

Different Types of Files

File Data Structure Definition Review

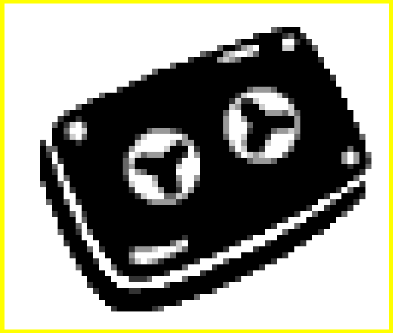
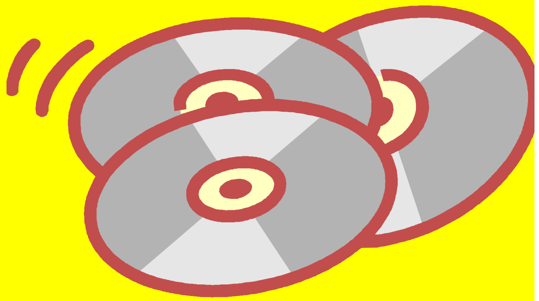
A *file* is an *internal* data structure – with an unspecified number of elements of the same type – assigned to an *external* file name.

The file data structure allows transfer of data between internal and external storage.



External File Definition

An *external file* is a sequence of information stored as bytes on a hard drive, disk, tape, CD, DVD, jump drive, or some other external storage device.

Text Files and Binary Files

Text files contain only characters and can be edited by any text editor like **Notepad** or **jGRASP**.

All Python programs are text files.

Binary files have a specific format which requires a specific application for editing.

2 Ways to Classify Files

- **by what they store** (*text* vs. *binary*)
- **by how the information is accessed** (*sequential access* vs. *random access*)

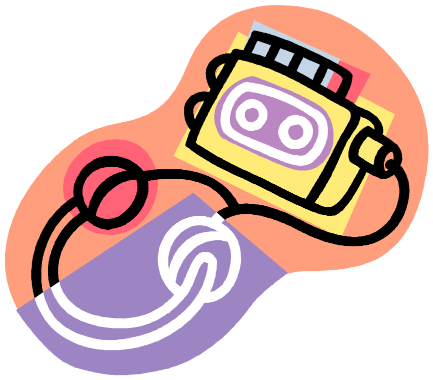
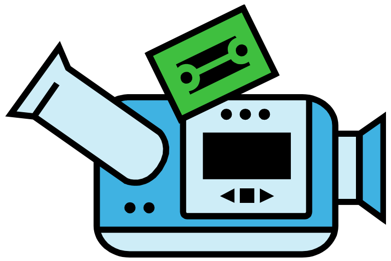
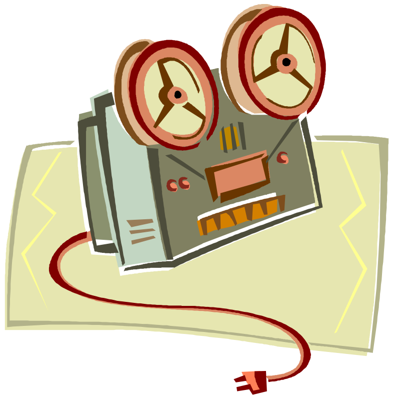
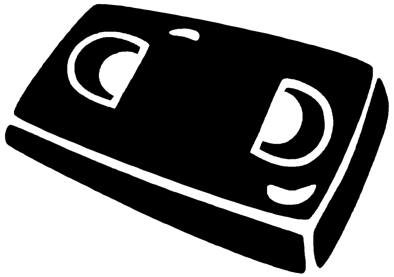
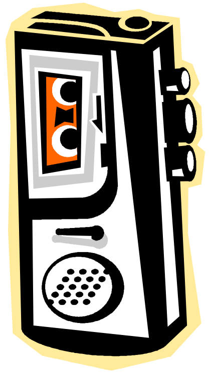
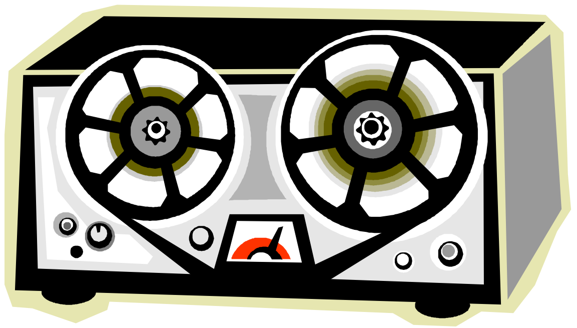
Sequential Access and Random Access

Files can have *sequential access* or *random access*.

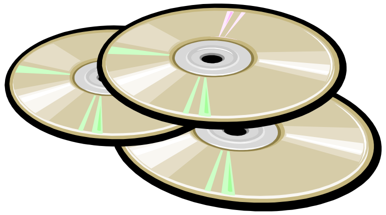
Sequential access files allow data access only in the sequence that the data is stored in the file.

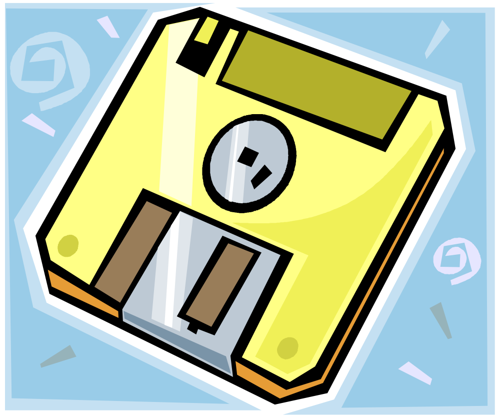
Random access files allow data access in any random pattern, regardless of how the data is stored.

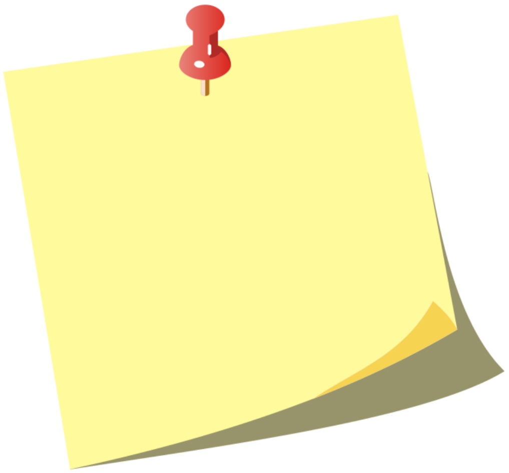
Sequential Access Examples

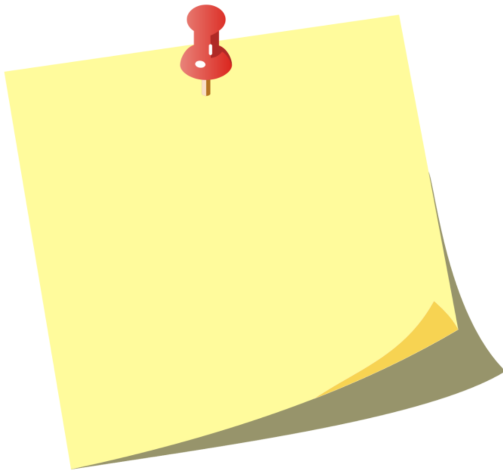
Random Access Examples







NOTE:



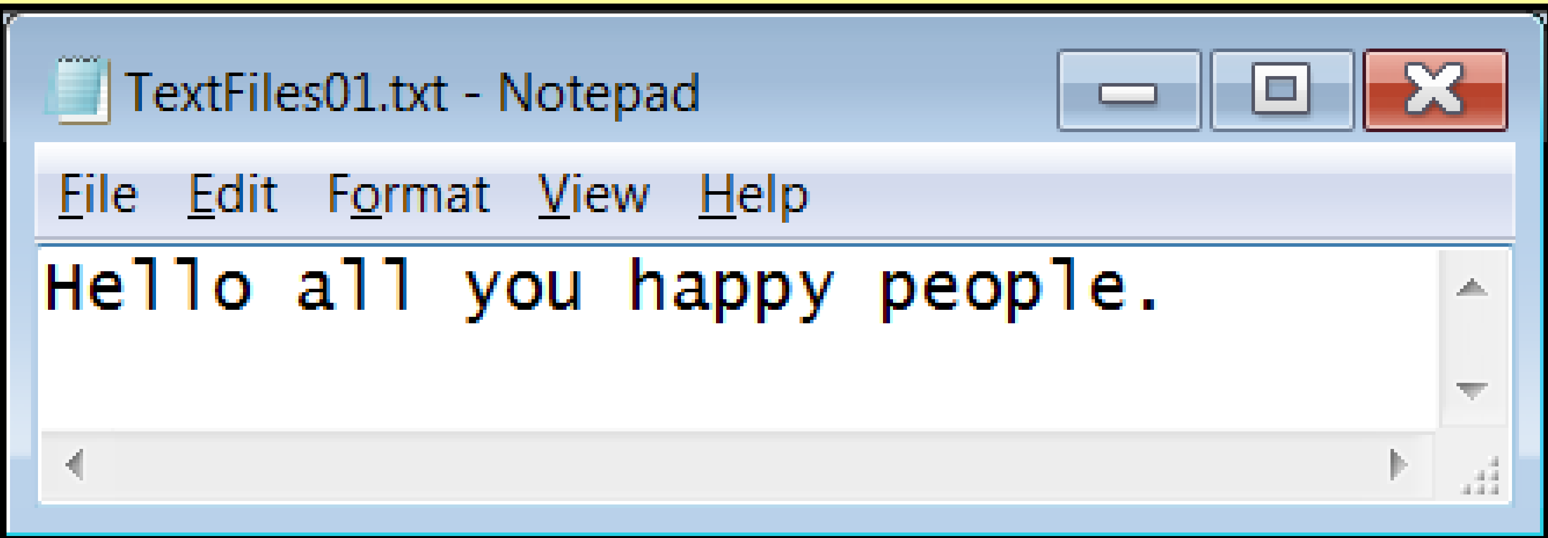
In this chapter,
we will strictly be working with
Sequential Text Files.

Section 17.3

Writing To and Reading From Text Files

```python
# TextFiles01.py
# This program demonstrates how to create a
# new text file and <write> some text to it.


file = open("TextFiles01.txt",'w')

file.write("Hello all you happy people.")

file.close()
```

```python
# TextFiles01.py
# This program demonstrates how to create a
# new text file and <write> some text to it.


file = open("TextFiles01.txt",'w')

file.write("Hello all you happy people.")

file.close()
```

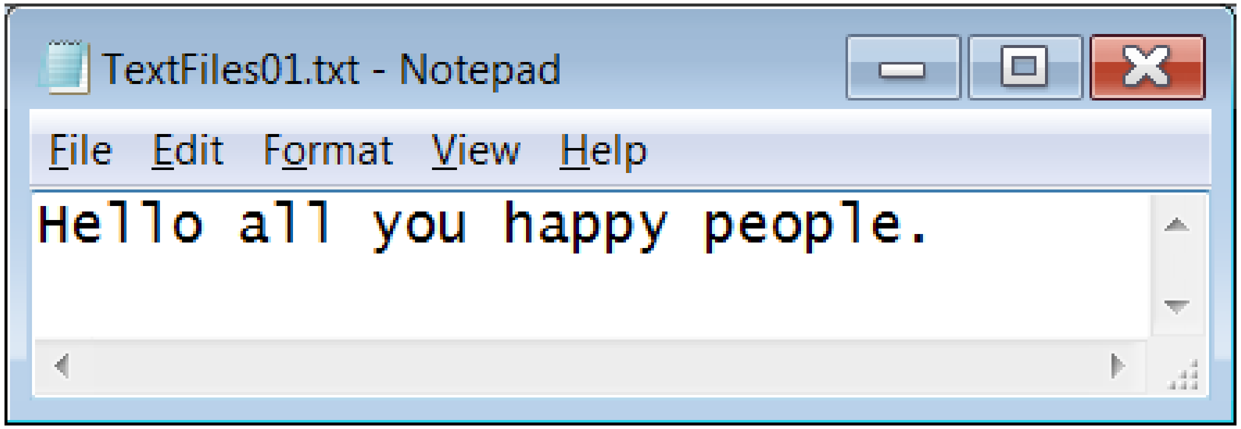


```
Hello all you happy people.
```

Computer Science Commandment

Anything that thou open, thou shalt also close!

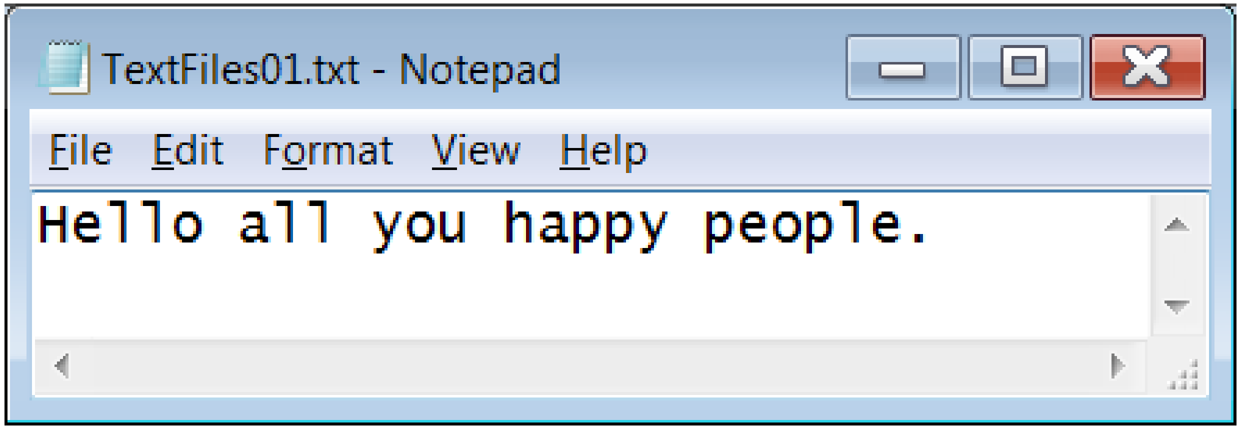
```python
# TextFiles02.py
# This program demonstrates how to <read> the text
# from the file created by the previous program.
# NOTE: If the previous program, TextFiles01.py,
# has not been executed, this program will crash.


file = open("TextFiles01.txt",'r')

text = file.readline()

print()
print(text)

file.close()

```



```
Hello all you happy people.
```

```
 ----jGRASP exec: python TextFiles02

Hello all you happy people.

 ----jGRASP: operation complete.
```

```python

file = open("TextFiles01.txt",'r')

text = file.readline()

print()
print(text)

file.close()

```



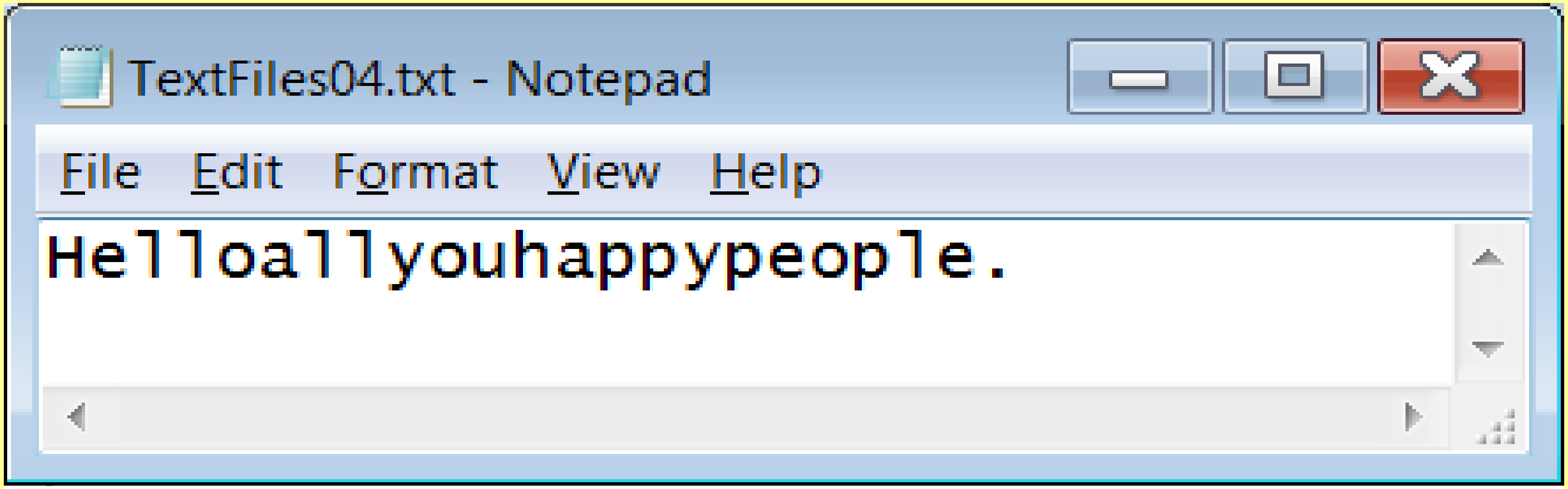
```
Hello all you happy people.
```

```python
# TextFiles03.py
# This program demonstrates a way to check
# if a file <exists> and also a way to see
# how many bytes a file contains.


import os


print()
if os.path.exists("TextFiles01.txt"):
    print("TextFiles01.txt stores",
          os.path.getsize("TextFiles01.txt"),
          "bytes of data.")
else:
    print("TextFiles01.txt does not exist.")

print()
if os.path.exists("qwerty.txt"):
    print("qwerty.txt stores",
          os.path.getsize("qwerty.txt"),
          "bytes of data.")
else:
    print("qwerty.txt does not exist.")
```

```
 ----jGRASP exec: python TextFiles03.py

TextFiles01.txt stores 27 bytes of data.

qwerty.txt does not exist.

 ----jGRASP: operation complete.
```

```python
if os.path.exists("TextFiles01.txt"):
   print("TextFiles01.txt stores",
         os.path.getsize("TextFiles01.txt"),
         "bytes of data.")
else:
   print("TextFiles01.txt does not exist.")

print()
if os.path.exists("qwerty.txt"):
   print("qwerty.txt stores",
         os.path.getsize("qwerty.txt"),
         "bytes of data.")
else:
   print("qwerty.txt does not exist.")
```

```python
# TextFiles04.py
# This program attempts to send multiple lines
# of output to a text file.  After you run this
# program, load the file TextFiles04.txt and
# you will see this did not work properly.


file = open("TextFiles04.txt",'w')

file.write("Hello")
file.write("all")
file.write("you")
file.write("happy")
file.write("people.")

file.close()
```

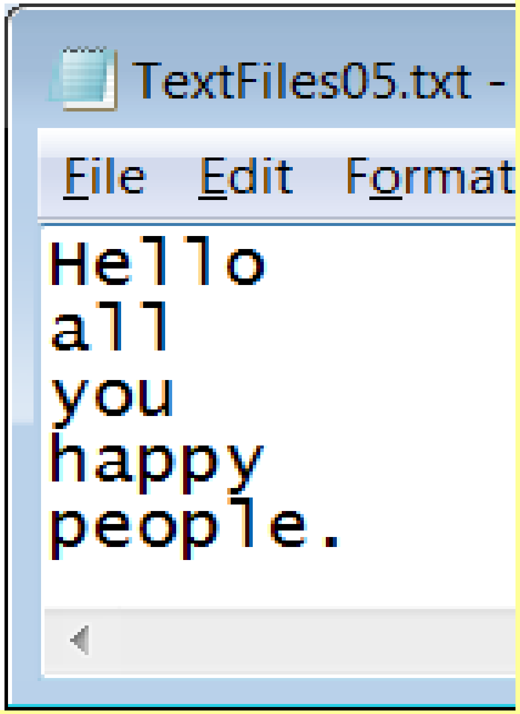


```
Helloallyouhappypeople.
```

```python
file = open("TextFiles04.txt",'w')

file.write("Hello")
file.write("all")
file.write("you")
file.write("happy")
file.write("people.")

file.close()
```

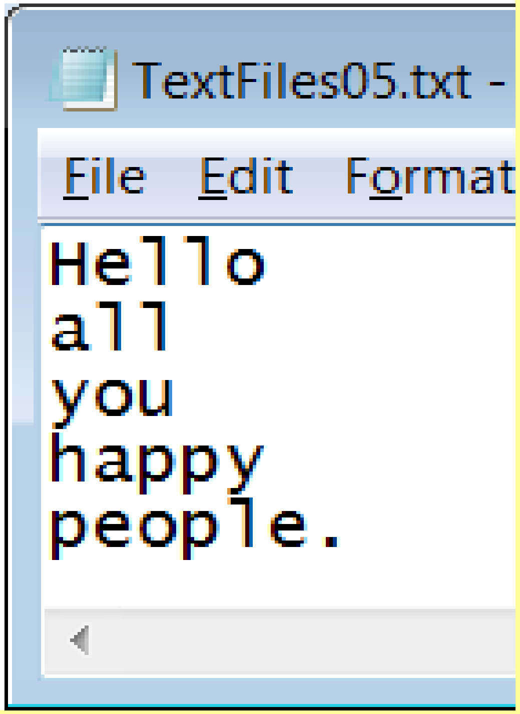
```python
# TextFiles05.py
# This program fixes the issue of the
# previous program by adding the
# "new line escape sequence" <\n>.


file = open("TextFiles05.txt",'w')

file.write("Hello\n")
file.write("all\n")
file.write("you\n")
file.write("happy\n")
file.write("people.\n")

file.close()
```

```python
# TextFiles05.py
# This program fixes the issue of the
# previous program by adding the
# "new line escape sequence" <\n>.


file = open("TextFiles05.txt",'w')

file.write("Hello\n")
file.write("all\n")
file.write("you\n")
file.write("happy\n")
file.write("people.\n")

file.close()
```



```
Hello
all
you
happy
people.
```

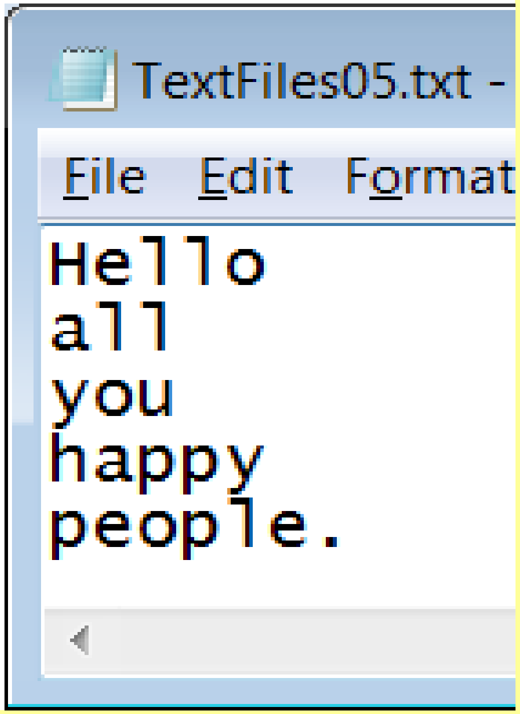
```python
# TextFiles06.py
# This program tries to read the file created
# by the previous program.  The problem is only
# one <readline> command is executed, so only
# one line of text is read and displayed.
# NOTE: If the previous program, TextFiles05.py,
#       has not been executed, this program, and
#       the next several programs, will crash.


file = open("TextFiles05.txt",'r')

lineOfText = file.readline()

print()
print(lineOfText)

file.close()
```



```
Hello
all
you
happy
people.
```

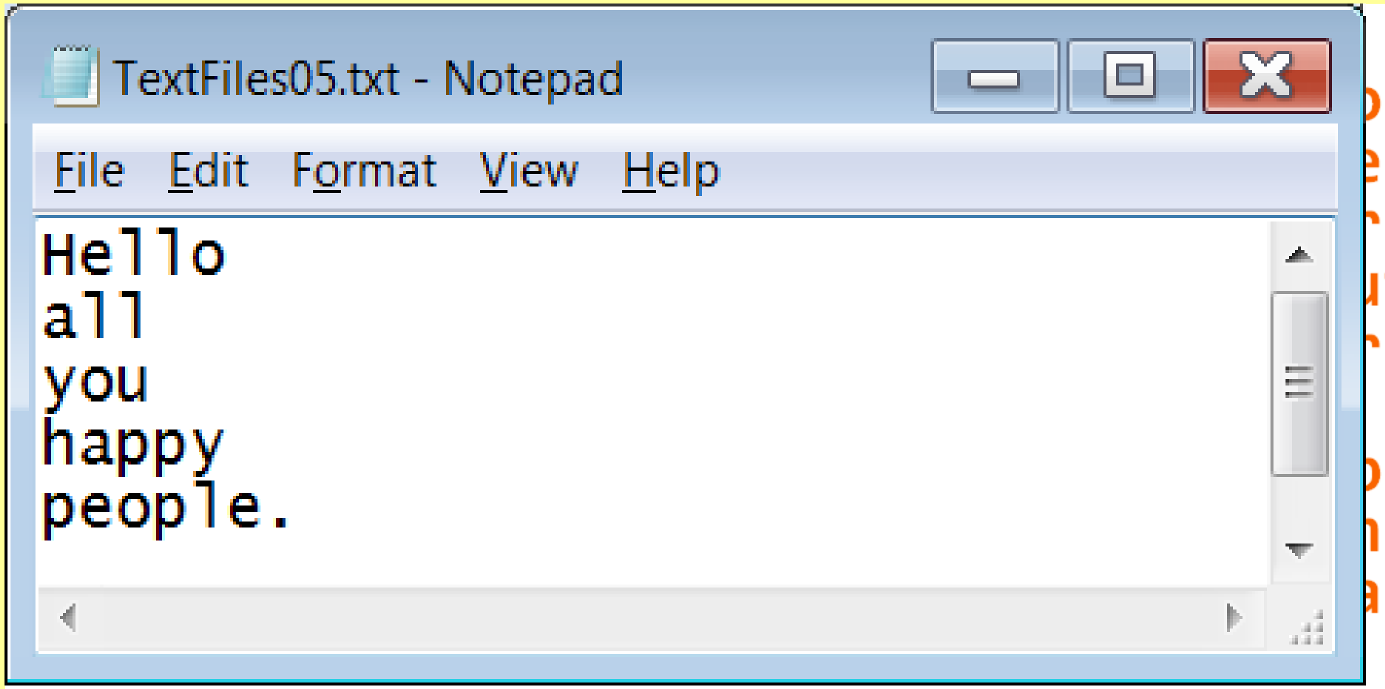
```
 ----jGRASP exec: python TextFiles06.py

Hello


 ----jGRASP: operation complete.
```

```


file = open("TextFiles05.txt",'r')

lineOfText = file.readline()

print()
print(lineOfText)

file.close()
```



```
Hello
all
you
happy
people.
```

```python
# TextFiles07.py
# This program uses a <for> loop to read
# all 5 lines of text from the file.
# There are a couple problems.  First, there
# are extra lines skipped in the output, caused
# by the <\n> characters used to write the
# data when the file was created.
# Second, this program only works because we
# know, ahead of time, how many lines of text
# are in the file, which is not realistic.


file = open("TextFiles05.txt",'r')

print()

for k in range(5):
    lineOfText = file.readline()
    print(lineOfText)

file.close()
```

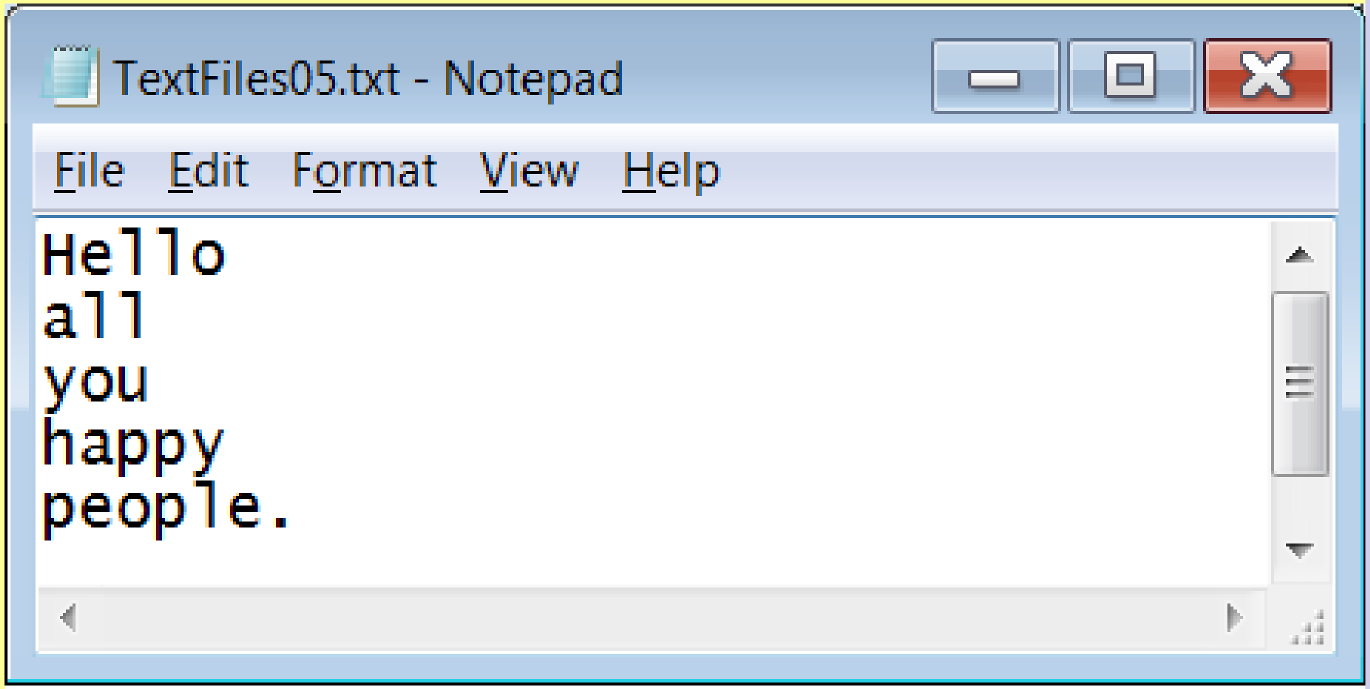


```
Hello
all
you
happy
people.
```

o read
e.
rst, there
utput, caused
rite the

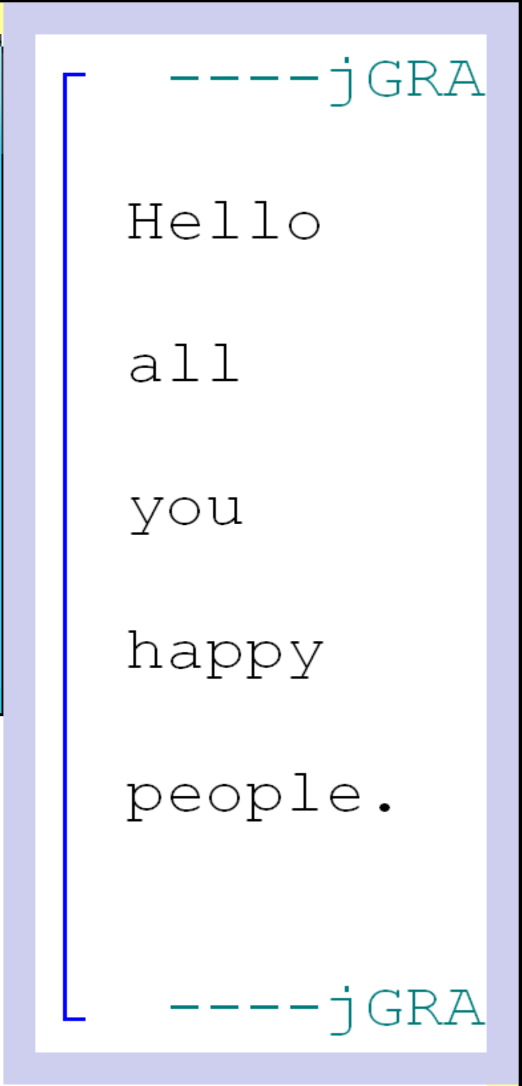
because we
nes of text
alistic.

```

file = open("TextFiles05.txt",'r')

print()

for k in range(5):
    lineOfText = file.readline()
    print(lineOfText)

file.close()
```



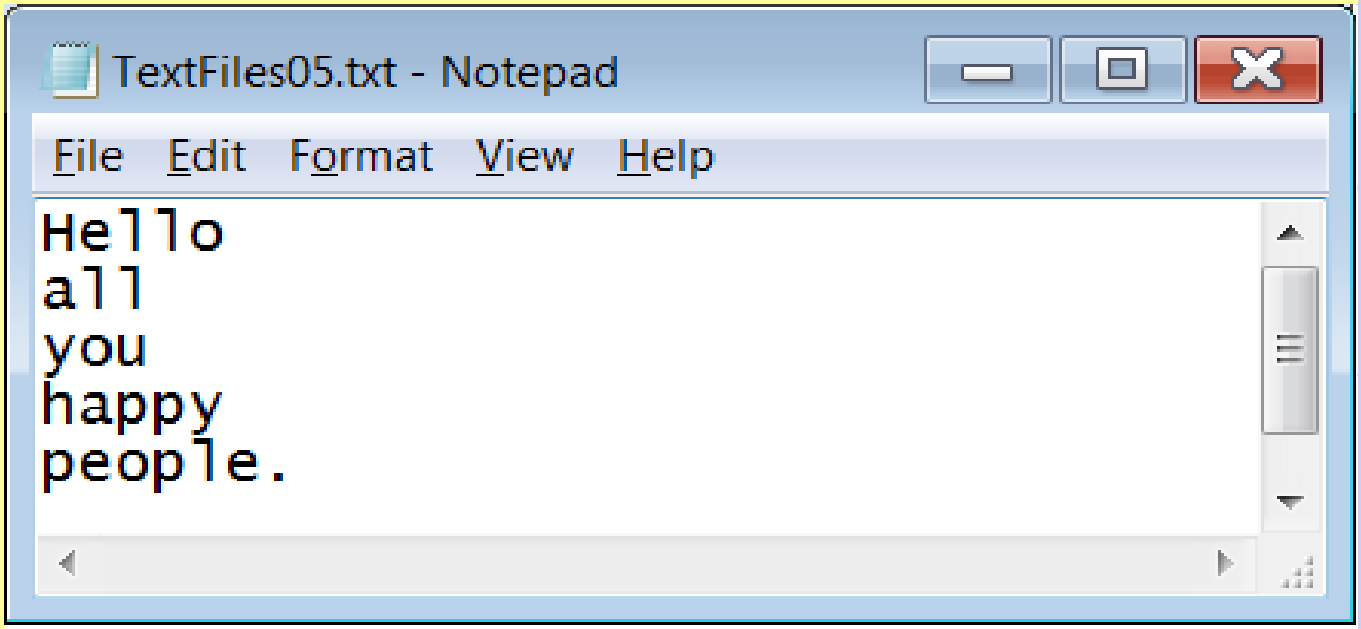
```
Hello
all
you
happy
people.
```

```python
file = open("TextFiles05.txt",'r')

print()

for k in range(5):
    lineOfText = file.readline()
    print(lineOfText)

file.close()
```



```
----jGRA

Hello

all

you

happy

people.


----jGRA
```

```python
# TextFiles08.py
# This program fixes the first issue of the
# previous program by using the <strip> command
# to "strip" away any invisible characters
# before and after each line of text.
# This includes </n> characters.


file = open("TextFiles05.txt",'r')

print()

for k in range(5):
   lineOfText = file.readline().strip()
   print(lineOfText)

file.close()
```

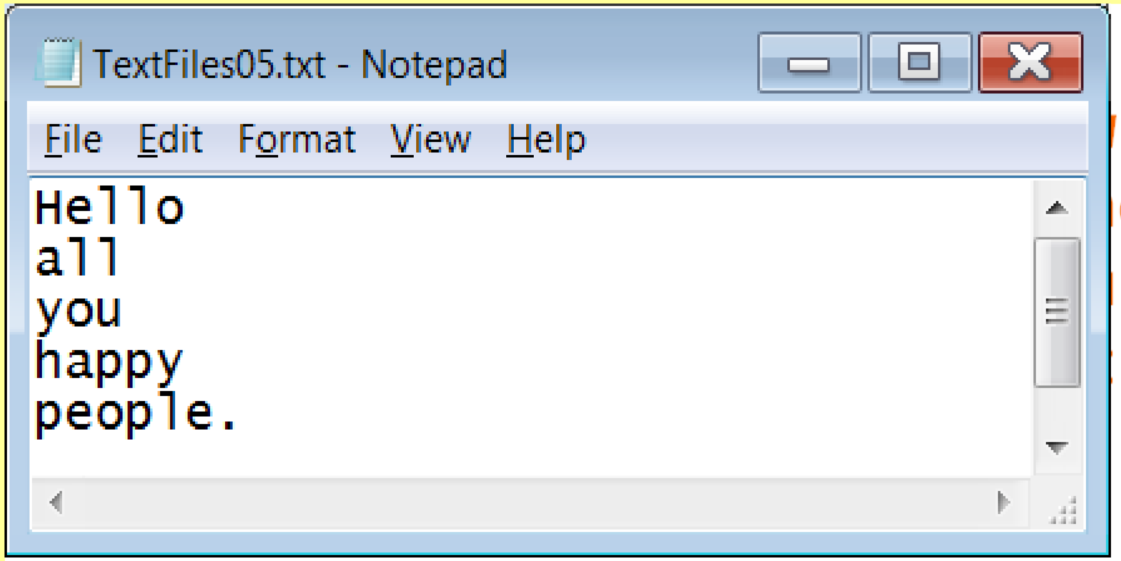


```
Hello
all
you
happy
people.
```

```python
file = open("TextFiles05.txt",'r')

print()

for k in range(5):
    lineOfText = file.readline().strip()
    print(lineOfText)

file.close()
```

```
    ----jG

 Hello
 all
 you
 happy
 people.

    ----jG
```

```python
# TextFiles09.py
# This program demonstrates what happens when you
# attempt to read past the end of a text file.
# In most languages, this would crash the program.
# In Python, you just get extra blank lines.


file = open("TextFiles05.txt",'r')

print()

for k in range(8):
   lineOfText = file.readline().strip()
   print(lineOfText)

file.close()

```



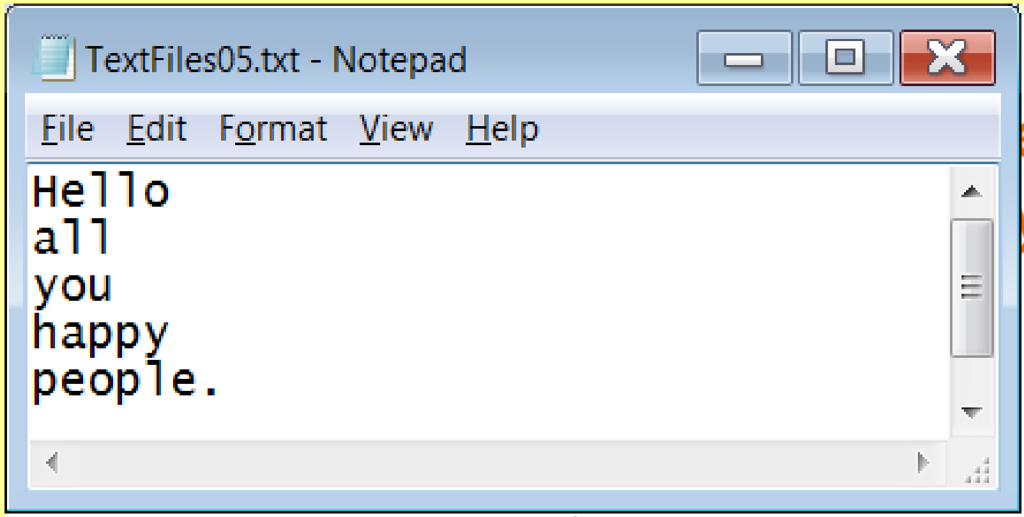
```
Hello
all
you
happy
people.
```

hat happe
d of a te
ld crash
ra blank

```
file = open("TextFiles05.txt",'r')

print()

for k in range(8):
    lineOfText = file.readline().strip()
    print(lineOfText)

file.close()

```

```
----jG

Hello
all
you
happy
people.



----jG
```

```python
# TextFiles10.py
# This program fixes the second issue from
# program TextFiles07.py by using the same
# <for..each> loop that works with arrays.


file = open("TextFiles05.txt",'r')

print()

for lineOfText in file:
    lineOfText = lineOfText.strip()
    print(lineOfText)

file.close()

```



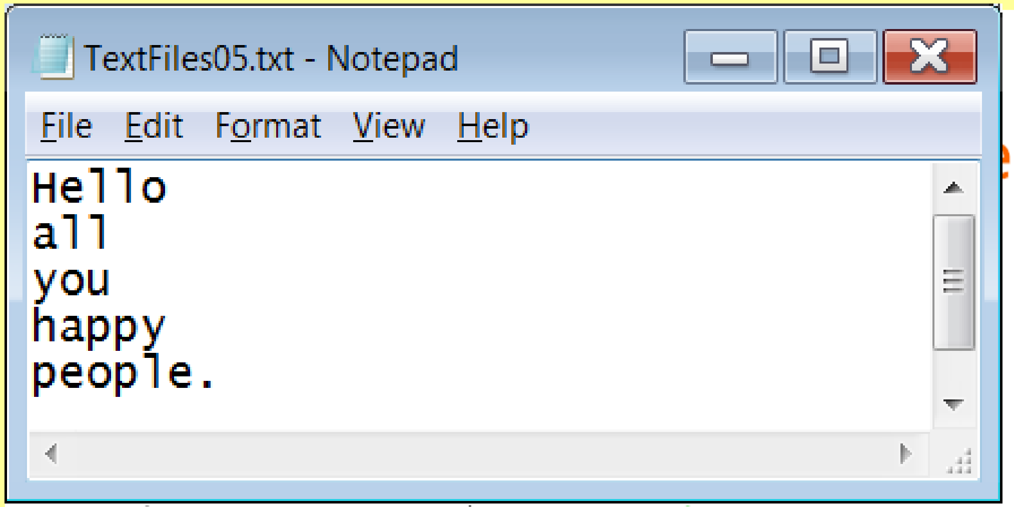
```
Hello
all
you
happy
people.
```

e second iss
y by using t
works with

```python
file = open("TextFiles05.txt",'r')

print()

for lineOfText in file:
    lineOfText = lineOfText.strip()
    print(lineOfText)

file.close()

```

```
----jG

Hello
all
you
happy
people.

----jG
```

```python
# TextFiles11.py
# This program demonstrates the <readlines> command which
# reads in the ENTIRE file and creates an array of strings.


file = open("TextFiles05.txt",'r')
allLinesOfText = file.readlines()
file.close()

print()
for lineOfText in allLinesOfText:
   lineOfText = lineOfText.strip()
   print(lineOfText)

```



```
Hello
all
you
happy
people.
```

<readlines> co

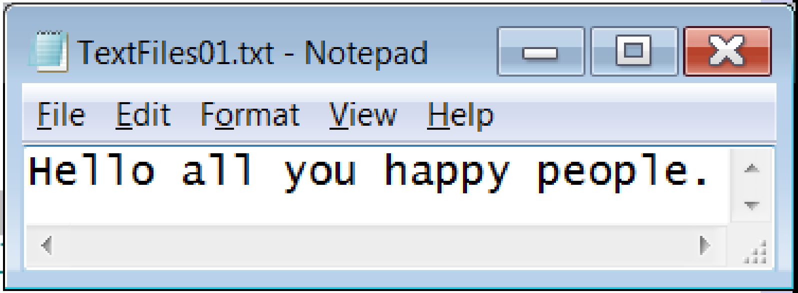
creates an arra

```python
file = open("TextFiles05.txt",'r')
allLinesOfText = file.readlines()
file.close()

print()
for lineOfText in allLinesOfText:
    lineOfText = lineOfText.strip()
    print(lineOfText)

```

```
----jG

Hello
all
you
happy
people.

----jG
```

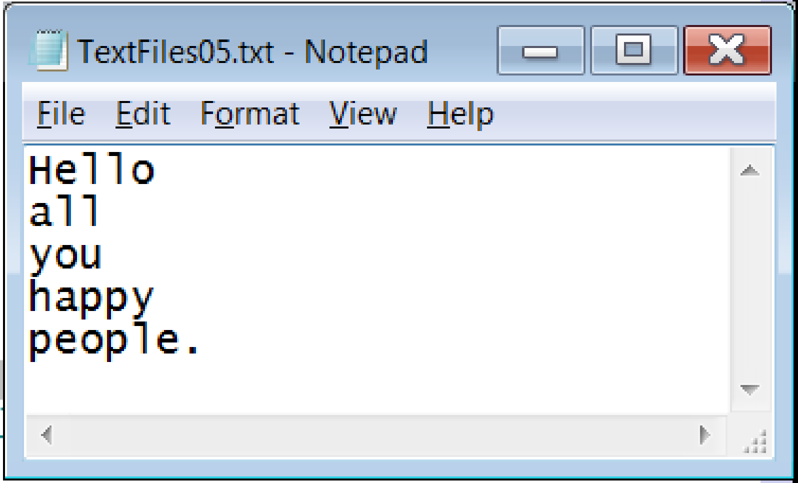
```python
# TextFiles12.py
# This program allows the user to enter a file name.
# Then it displays the contents of that file,
# provided it exists.


import os


print()
fileName = input("Enter the name of a file.  -->  ")
print()

if os.path.exists(fileName):
   file = open(fileName,'r')
   for text in file:
      text = text.strip()
      print(text)
   file.close()
else:
   print(fileName,"does not exist.")
```

```
 ----jGRASP exec: python TextFiles12.py

Enter the name of a file.  -->  TextFiles01.txt

Hello all you happy people.

 ----jGRASP: operation complete
```



```
Hello all you happy people.
```

```
 ----jGRASP exec: python TextFi

Enter the name of a file.  -->  TextFiles05.txt

Hello
all
you
happy
people.

 ----jGRASP: operation complete
```



```
Hello
all
you
happy
people.
```

```
 ----jGRASP exec: python TextFi

Enter the name of a file.  -->  qwerty.txt

qwerty.txt does not exist.

 ----jGRASP: operation complete.
 ----jGRASP exec: python TextFiles12.py

Enter the name of a file.  -->  TextFiles12.py

# TextFiles12.py
# This program allows the user to enter a file name.
# Then it displays the contents of that file,
# provided it exists.


import os



print()
fileName = input("Enter the name of a file.  -->  ")
print()

if os.path.exists(fileName):
file = open(fileName,'r')
for text in file:
text = text.strip()
print(text)
file.close()
else:
print(fileName,"does not exist.")

 ----jGRASP: operation complete.
 ----jGRASP exec: python TextFiles12.py

Enter the name of a file.  -->  TextFiles12.py

# TextFiles12.py
# This program allows the user to enter a file name.
# Then it displays the contents of that file,
# provided it exists.


import os


print()
fileName = input("Enter the name of a file.  -->  ")
print()

if os.path.exists(fileName):
file = open(fileName,'r')
for text in file:
text = text.strip()
print(text)
file.close()
else:
print(fileName,"does not exist.")

 ----jGRASP: operation complete.
```

Yes, this program is actually displaying itself, but what happened to the indenting?

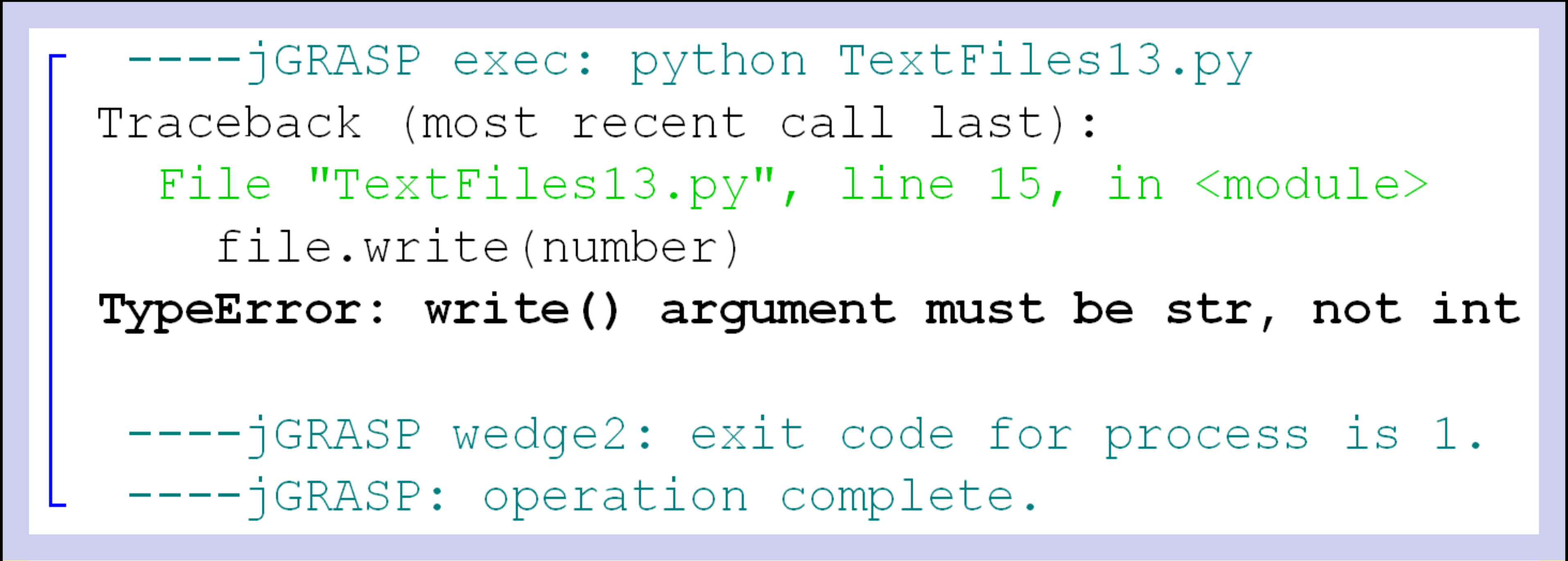
```
 ----jGRASP exec: python TextFiles12.py

Enter the name of a file.  -->  TextFiles12.py

# TextFiles12.py
# This program allows the user to enter a file name.
# Then it displays the contents of that file,
# provided it exists.


import os



print()
fileName = input("Enter the name of a file.  -->  ")
print()

if os.path.exists(fileName):
file = open(fileName,'r')
for text in file:

print(text)
file.close()
else:
print(fileName,"does not exist.")

 ----jGRASP: operation complete.
```

Yes, this program is actually displaying itself, but what happened to the indenting?

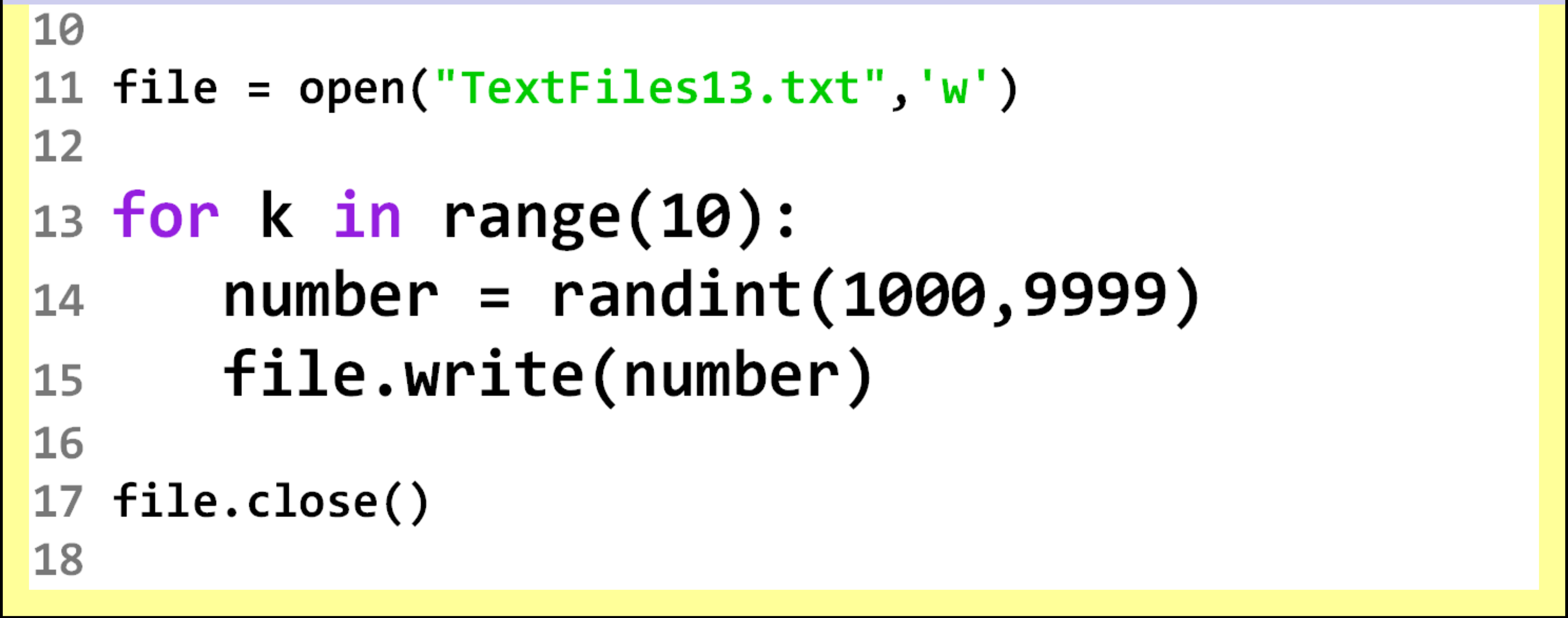
Section 17.4

Files of Numbers

```python
# TextFiles13.py
# This program attempts to create a file of 10 random
# integers.  The program has a syntax error because
# only string values can be written to a text file.


from random import *

seed(1234)

file = open("TextFiles13.txt",'w')

for k in range(10):
    number = randint(1000,9999)
    file.write(number)

file.close()

```



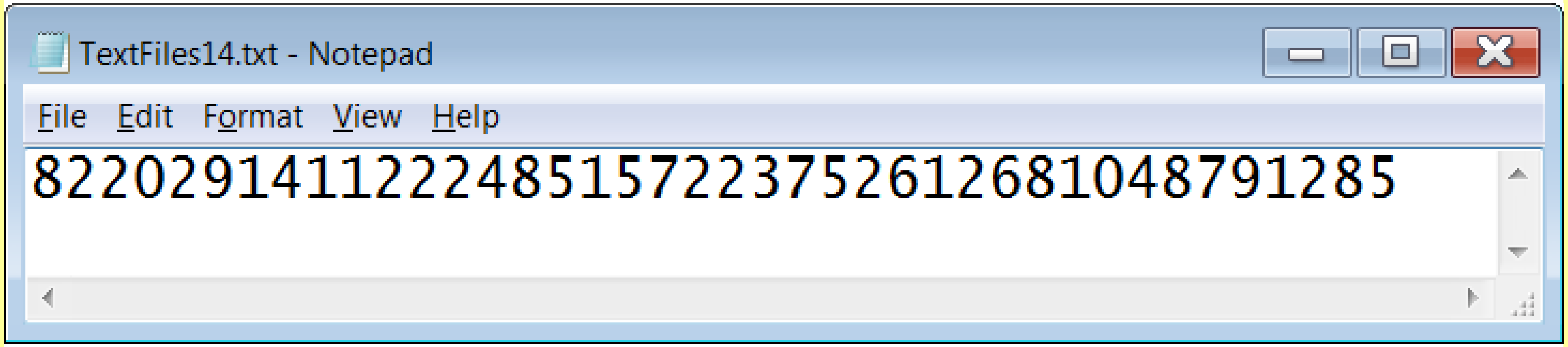
```
 ----jGRASP exec: python TextFiles13.py
Traceback (most recent call last):
  File "TextFiles13.py", line 15, in <module>
    file.write(number)
TypeError: write() argument must be str, not int

 ----jGRASP wedge2: exit code for process is 1.
 ----jGRASP: operation complete.
```



```
10
11 file = open("TextFiles13.txt",'w')
12
13 for k in range(10):
14     number = randint(1000,9999)
15     file.write(number)
16
17 file.close()
18
```

```
# TextFiles14.py
# This program fixes the issue of the previous program
# by using the <str> command to convert the numbers to
# strings.  However, a new issue is created; which can
# be seen by loading the file TextFiles15.txt


from random import *

seed(1234)

file = open("TextFiles14.txt",'w')

for k in range(10):
   number = randint(1000,9999)
   file.write(str(number))

file.close()

```

```python
# TextFiles14.py
```

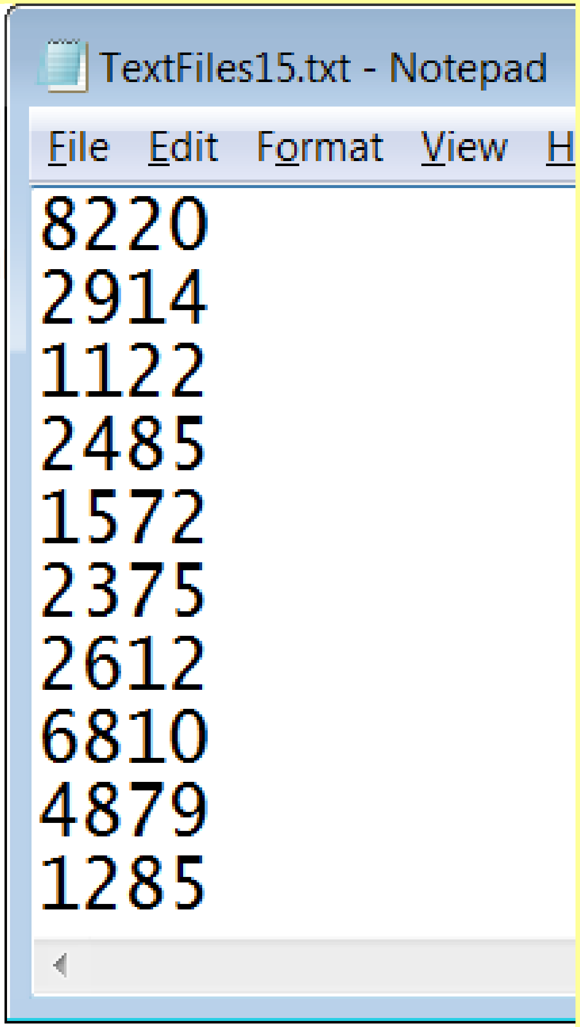


```
8220291411222485157223752612681048791285
```

```python
from random import *

seed(1234)

file = open("TextFiles14.txt",'w')

for k in range(10):
    number = randint(1000,9999)
    file.write(str(number))

file.close()

```

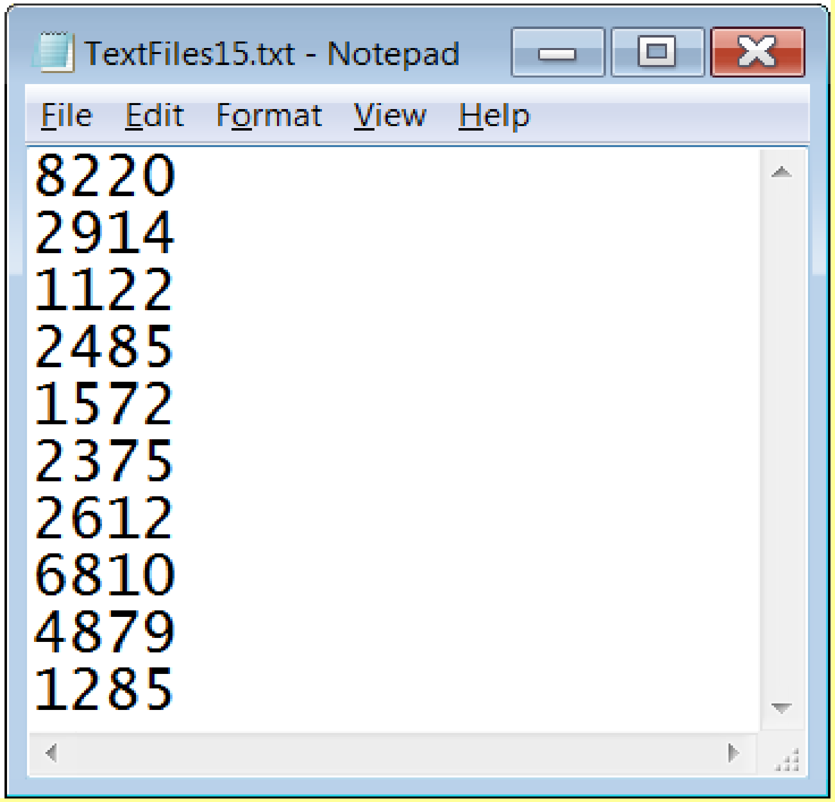
```python
# TextFiles15.py
# This program fixes the issue of the previous
# program by adding a "new line escape sequence" <\n>
# as was done in earlier program examples.


from random import *

seed(1234)

file = open("TextFiles15.txt",'w')

for k in range(10):
    number = randint(1000,9999)
    file.write(str(number)+"\n")

file.close()

```

```python
# TextFiles15.py
# This program fixes the issue of th
# program by adding a "new line esca
# as was done in earlier program exa


from random import *

seed(1234)

file = open("TextFiles15.txt",'w')

for k in range(10):
    number = randint(1000,9999)
    file.write(str(number)+"\n")

file.close()

```



```
8220
2914
1122
2485
1572
2375
2612
6810
4879
1285
```

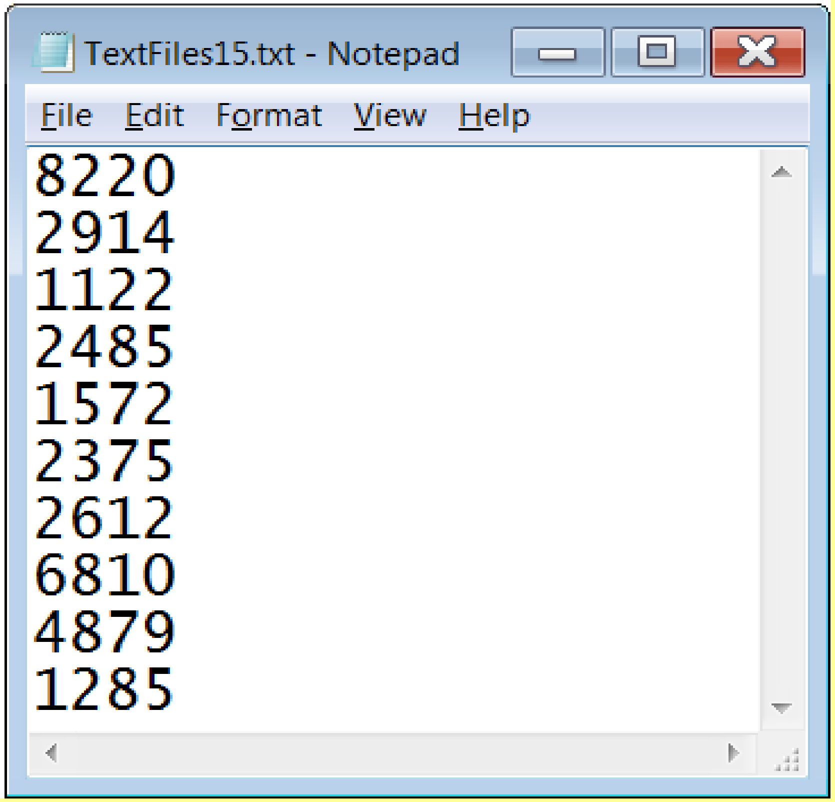
```python
# TextFiles16.py
# This program attempts to read in the numbers from the
# file created by the previous program and average them.
# The problem is the numbers are stored as text and need to
# be converted back to numbers before they can be averaged.


file = open("TextFiles15.txt",'r')

print()

total = 0
count = 0

for number in file:
    print(number)
    total += number
    count += 1

file.close()

average = total / count

print()
print("Total   =",total)
print()
print("Average =",average)
```



```
8220
2914
1122
2485
1572
2375
2612
6810
4879
1285
```

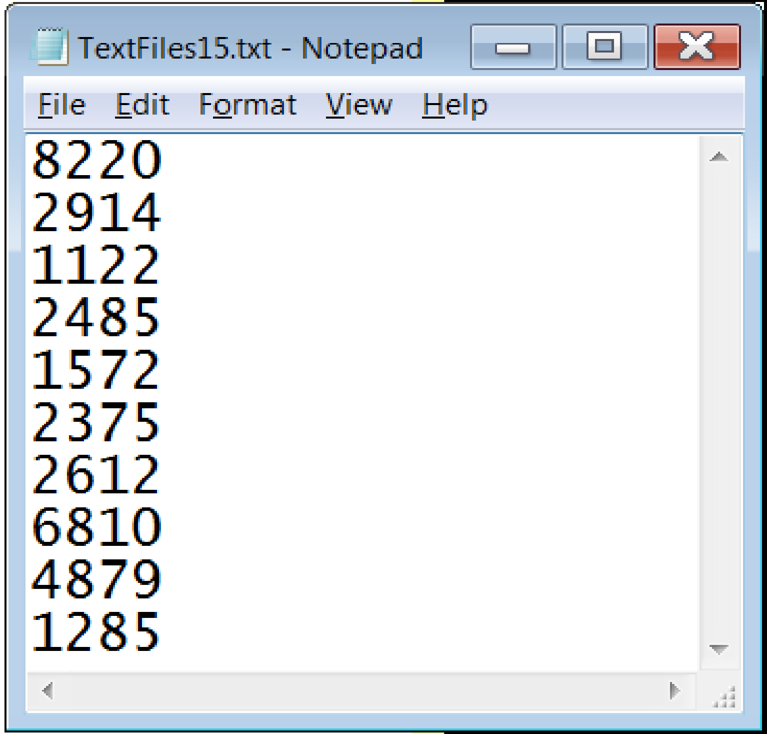
```
 ----jGRASP exec: python TextFiles16.py

8220

Traceback (most recent call last):
  File "TextFiles16.py", line 17, in <module>
    total += number
TypeError: unsupported operand type(s) for +=: 'int' and 'str'
```

```
12 total = 0
13 count = 0
14
15 for number in file:
16    print(number)
17    total += number
18    count += 1
19
20 file.close()
21
22 average = total / count
23
24 print()
25 print("Total   =",total)
26 print()
27 print("Average =",average)
```



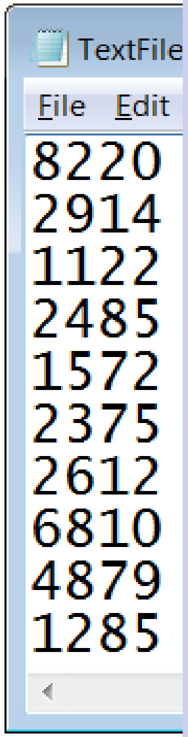
```
8220
2914
1122
2485
1572
2375
2612
6810
4879
1285
```

```python
# TextFiles17.py
# This program fixes the issue of the previous
# program by using the <int> command to convert
# the text numbers back into integer values.
# NOTE: This process also eliminates the <\n>
#       escape sequence character.


file = open("TextFiles15.txt",'r')
print()
total = 0
count = 0

for lineOfText in file:
    number = int(lineOfText)
    print(number)
    total += number
    count += 1

file.close()

average = total / count

print()
print("Total   =",total)
print()
print("Average =",average)
```

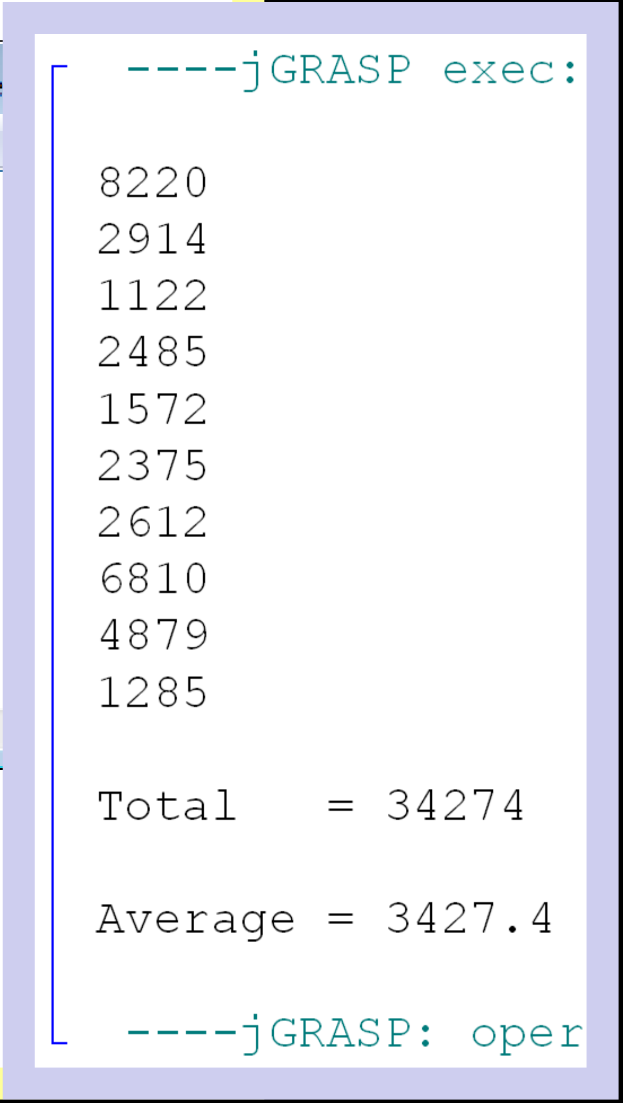


```
8220
2914
1122
2485
1572
2375
2612
6810
4879
1285
```

```python
# TextFiles17.py
# This program fixes the issue of the previous
# program by using the <int> command to convert
# the text numbers back into integer values.
# NOTE: This process also eliminates the <\n>
#        escape sequence character.


file = open("TextFiles15.txt",'r')
print()
total = 0
count = 0

for lineOfText in file:
    number = int(lineOfText)
    print(number)
    total += number
    count += 1

file.close()

average = total / count

print()
print("Total   =",total)
print()
print("Average =",average)
```



```
8220
2914
1122
2485
1572
2375
2612
6810
4879
1285
```



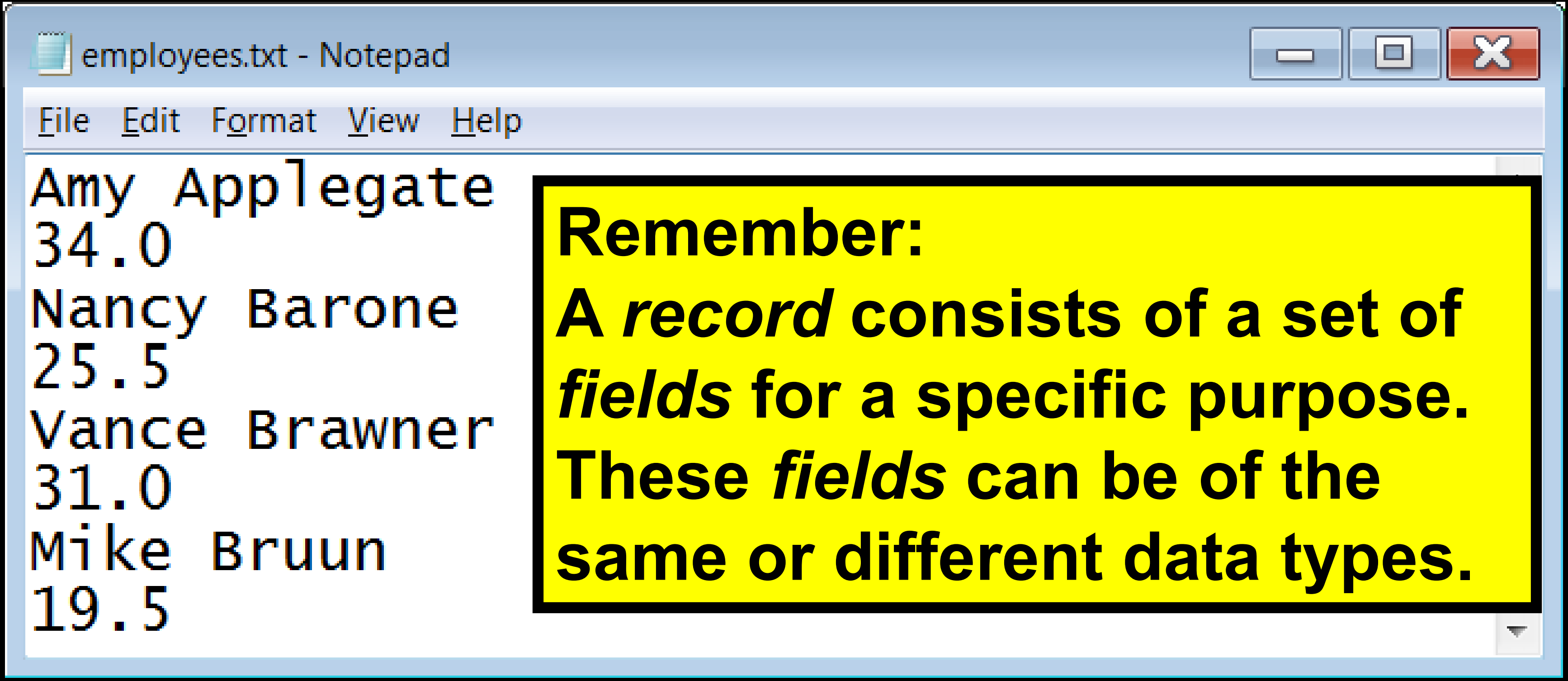
```
 ----jGRASP exec:

 8220
 2914
 1122
 2485
 1572
 2375
 2612
 6810
 4879
 1285

 Total   = 34274

 Average = 3427.4

 ----jGRASP: oper
```

Section 17.5

Files of Multiple Data Types

What Files Can Store

Files are not limited to merely storing either strings OR numbers. They can store BOTH strings AND numbers at the same time in the same file. This means that files can store *records* of information.

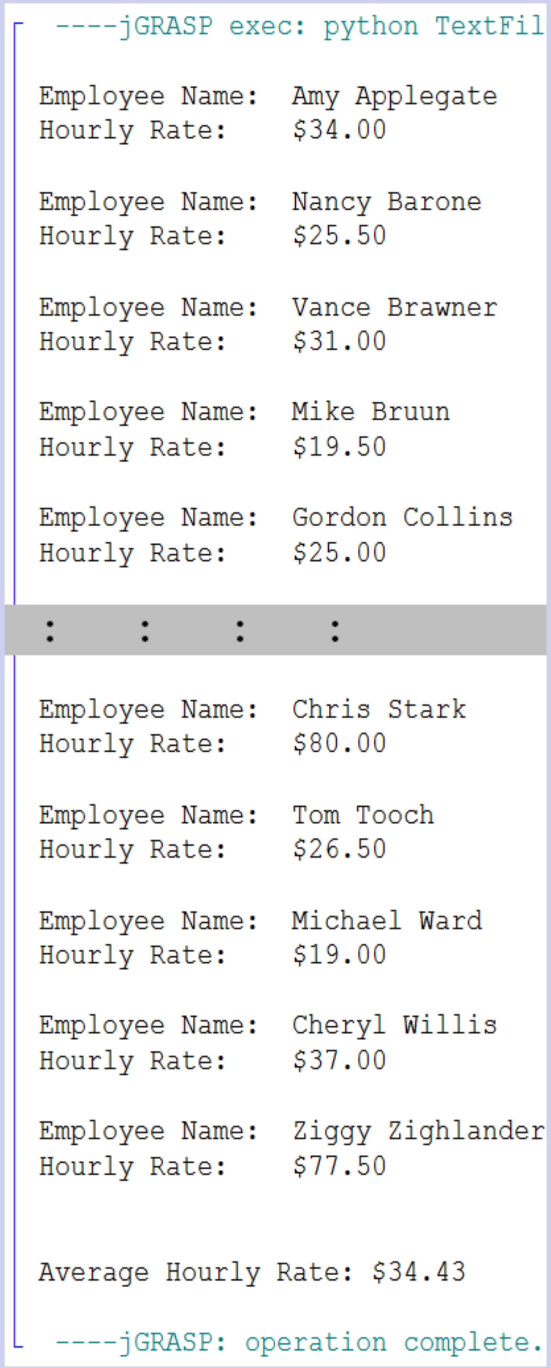


```
Amy Applegate
34.0
Nancy Barone
25.5
Vance Brawner
31.0
Mike Bruun
19.5
```

Remember:
A *record* consists of a set of *fields* for a specific purpose. These *fields* can be of the same or different data types.

```python
# TextFiles18.py
# This program demonstrates how to read information
# from a file containing different data types.
# In this case, the "employees.txt" file contains the
# names (string values) and hourly rates (real# values)
# for several employees.
# The file information is read into 2 "parallel arrays".
# After that, the information in both arrays is displayed,
# and the average hourly rate is computed and displayed.


file = open("employees.txt",'r')
names = []
hourlyRates = []
count = 0
for lineOfText in file:
    if (count % 2 == 0):
        names.append(lineOfText.strip())
    else:
        hourlyRates.append(float(lineOfText))
    count += 1
file.close()

count = 0
total = 0
for k in range(len(names)):
   print()
   print("Employee Name: ",names[k])
   print("Hourly Rate:   ","${:.2f}".format(hourlyRates[k]))
   total += hourlyRates[k]
   count += 1

average = total / count

print("\n")
print("Average Hourly Rate:","${:.2f}".format(average))
```

```python
# TextFiles18.py
# This program demonstrates how to read information
# from a file containing different data types.
# In this case, the "employees.txt" file contains the
# names (string values) and hourly rates (real# values)
# for several employees.
# The file information is read into 2 "parallel arrays".
# After that, the information in both arrays is displayed,
# and the average hourly rate is computed and displayed.


file = open("employees.txt",'r')
names = []
hourlyRates = []
count = 0
for lineOfText in file:
    if (count % 2 == 0):
        names.append(lineOfText.strip())
    else:
        hourlyRates.append(float(lineOfText))
    count += 1
file.close()

count = 0
total = 0
for k in range(len(names)):
   print()
   print("Employee Name: ",names[k])
   print("Hourly Rate:   ","${:.2f}".format(hourlyRates[k]))
   total += hourlyRates[k]
   count += 1

average = total / count

print("\n")
print("Average Hourly Rate:","${:.2f}".format(average))
```



```
 ----jGRASP exec: python TextFil

Employee Name:  Amy Applegate
Hourly Rate:    $34.00

Employee Name:  Nancy Barone
Hourly Rate:    $25.50

Employee Name:  Vance Brawner
Hourly Rate:    $31.00

Employee Name:  Mike Bruun
Hourly Rate:    $19.50

Employee Name:  Gordon Collins
Hourly Rate:    $25.00

  :    :    :    :

Employee Name:  Chris Stark
Hourly Rate:    $80.00

Employee Name:  Tom Tooch
Hourly Rate:    $26.50

Employee Name:  Michael Ward
Hourly Rate:    $19.00

Employee Name:  Cheryl Willis
Hourly Rate:    $37.00

Employee Name:  Ziggy Zighlander
Hourly Rate:    $77.50


Average Hourly Rate: $34.43

 ----jGRASP: operation complete.
```

Parallel Arrays Example

	names
0	Amy Applegate
1	Nancy Barone
2	Vance Brawner
3	Mike Bruun
4	Gordon Collions
:	:
35	Ziggy Zighlander

	hourlyRates
0	34.0
1	25.5
2	31.0
3	19.5
4	25.0
:	:
35	77.5

Parallel Arrays Disclaimer

When working with multiple records of information, the use of *Parallel Arrays* is not the only option, nor is it the best option.

It is much more organized, efficient, and even intuitive to use an *array of records* to store all of the information.

However, this requires a complexity in coding that we wish to avoid in a first year computer science class.

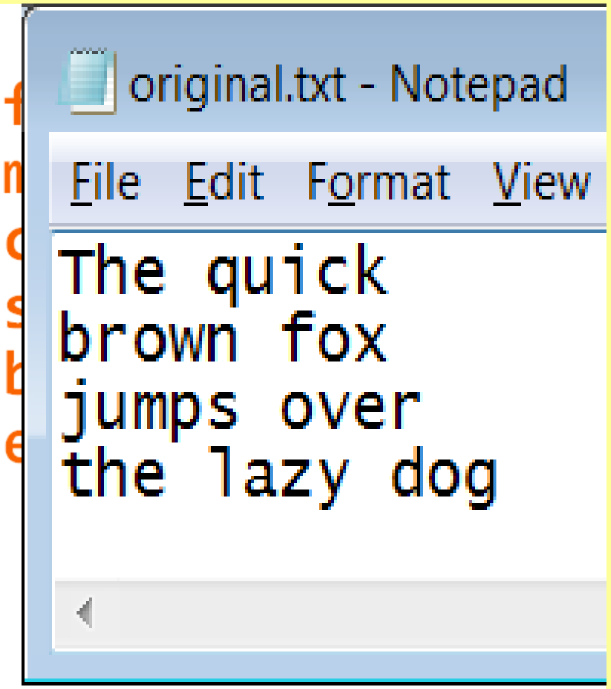
Arrays of Records will be discussed along with *Object Oriented Programming* in AP® Computer Science-A.

Section 17.6

Reading & Writing Simultaneously

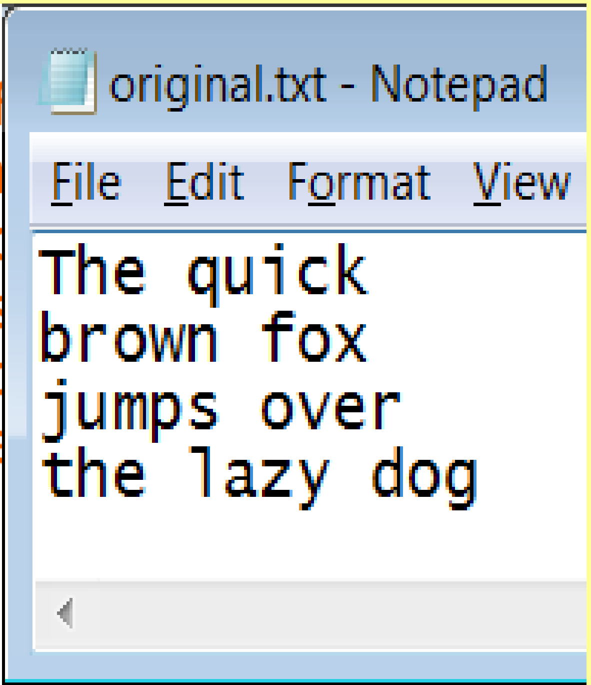
```python
# TextFiles19.py
# This program reads an original file and writes
# a backup file with the exact same contents.
# It demonstrates that a program can read from
# and/or write to multiple files simultaneously.
# After you run the program load both original.txt
# and backup.txt and you should see the two files
# are identical.


inFile = open("original.txt",'r')
outFile = open("backup.txt",'w')

print()

for lineOfText in inFile:
   outFile.write(lineOfText)

inFile.close()
outFile.close()
```

```
# TextFiles19.py
# This program reads an original f
# a backup file with the exact sam
# It demonstrates that a program c
# and/or write to multiple files s
# After you run the program load b
# and backup.txt and you should se
# are identical.


inFile = open("original.txt",'r')
outFile = open("backup.txt",'w')

print()

for lineOfText in inFile:
    outFile.write(lineOfText)

inFile.close()
outFile.close()
```

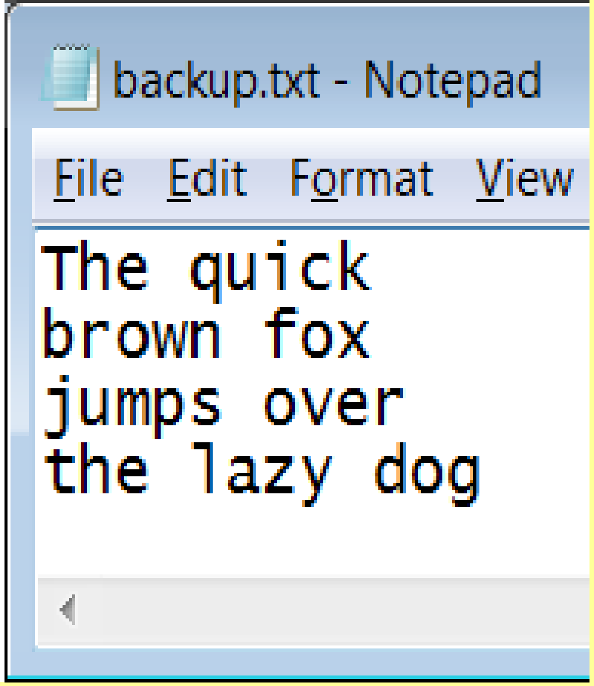


```
The quick
brown fox
jumps over
the lazy dog
```

```python
# TextFiles19.py
# This program reads an original f
# a backup file with the exact sam
# It demonstrates that a program c
# and/or write to multiple files s
# After you run the program load b
# and backup.txt and you should se
# are identical.


inFile = open("original.txt",'r')
outFile = open("backup.txt",'w')

print()

for lineOfText in inFile:
   outFile.write(lineOfText)

inFile.close()
outFile.close()
```



```
The quick
brown fox
jumps over
the lazy dog
```



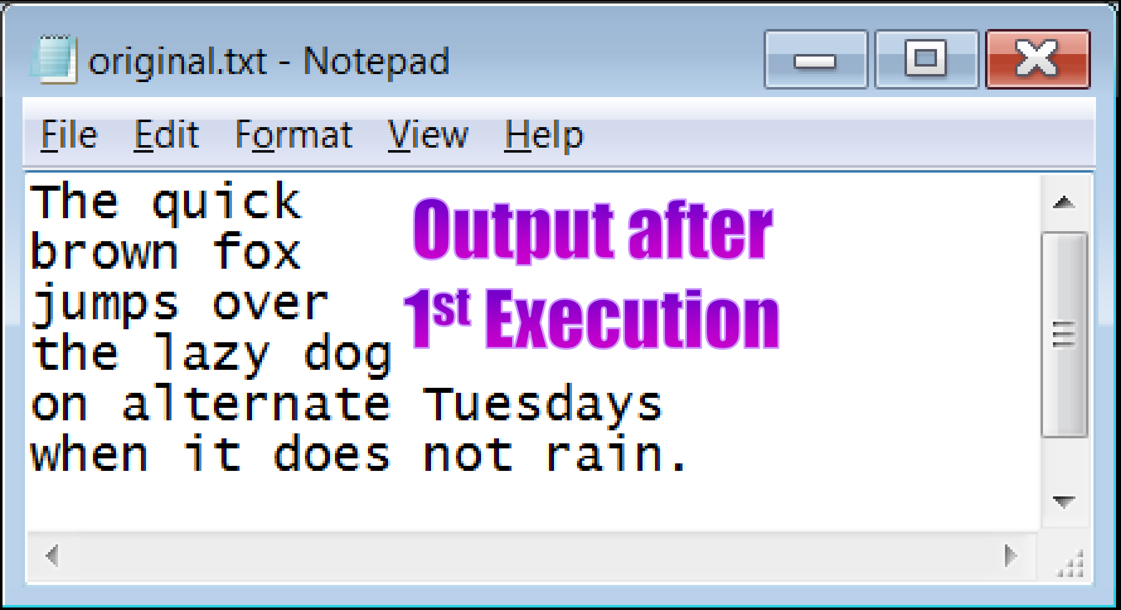
```
The quick
brown fox
jumps over
the lazy dog
```

Section 17.7

Appending to an Existing File

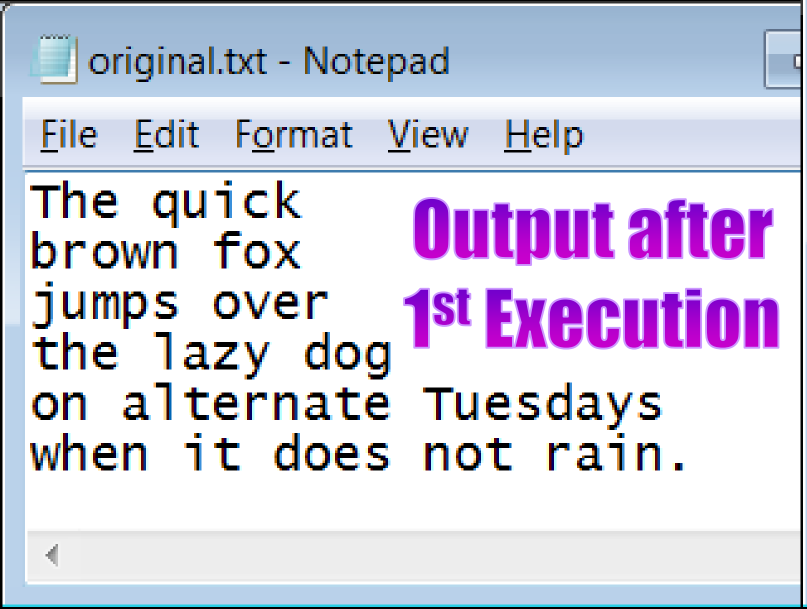
```python
# TextFiles20.py
# This program demonstrates how to "append"
# data to the end of an existing file.


file = open("original.txt",'a')

file.write("on alternate Tuesdays\n")
file.write("when it does not rain.\n")

file.close()
```

```python
# TextFiles20.py
# This program demonstrates how to "append"
# data to the end of an existing file.


file = open("original.txt",'a')

file.write("on alternate Tuesdays\n")
file.write("when it does not rain.\n")

file.close()
```



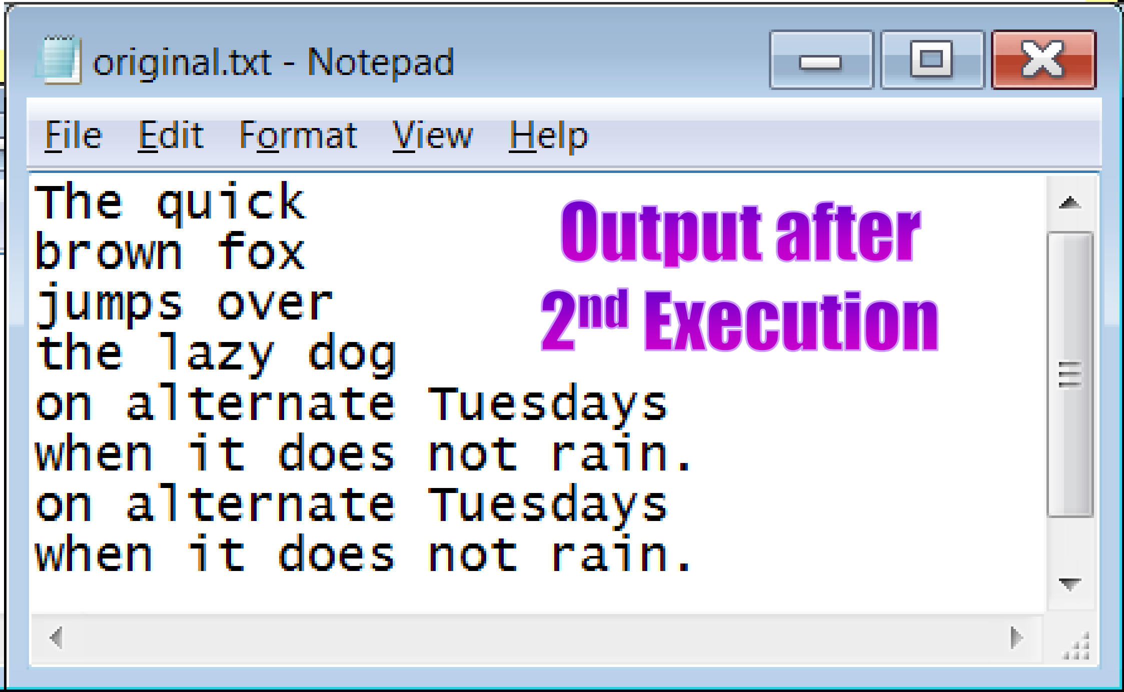
```
The quick
brown fox
jumps over
the lazy dog
on alternate Tuesdays
when it does not rain.
```

```python
# TextFiles20.py
# This program demonstrates how to "append"
# data to the end of an existing file.


file = open("original.txt",'a')

file.write("on alternate Tuesdays\n")
file.write("when it does not rain.\n")

file.close()
```



```
The quick
brown fox
jumps over
the lazy dog
on alternate Tuesdays
when it does not rain.
```

Output after 1st Execution



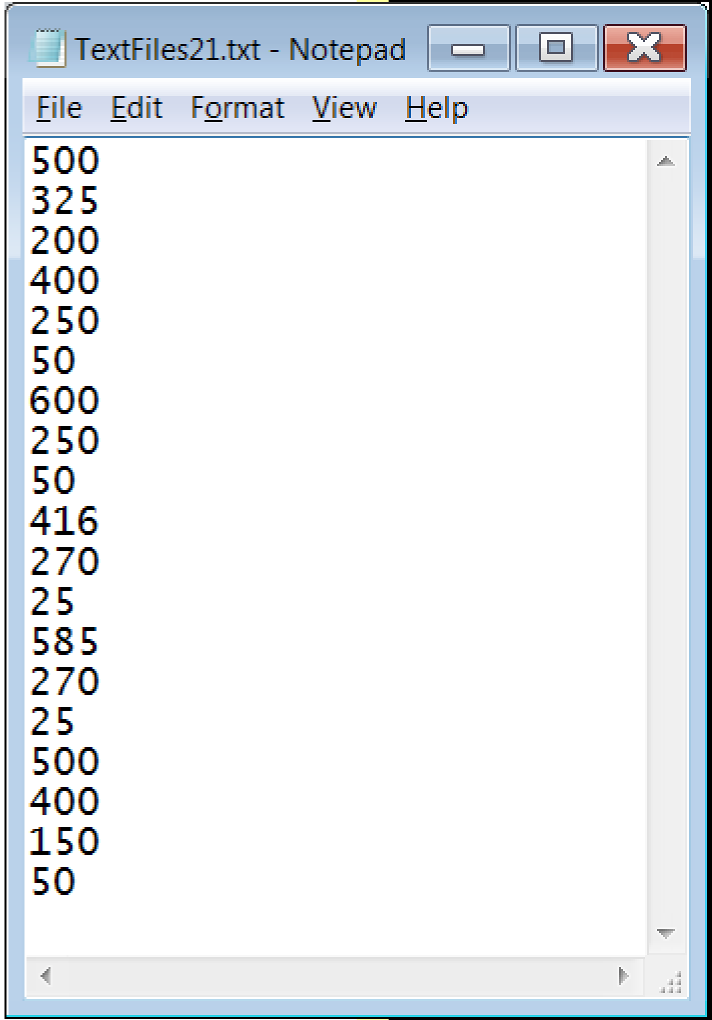
```
The quick
brown fox
jumps over
the lazy dog
on alternate Tuesdays
when it does not rain.
on alternate Tuesdays
when it does not rain.
```

Output after 2nd Execution

Section 17.8

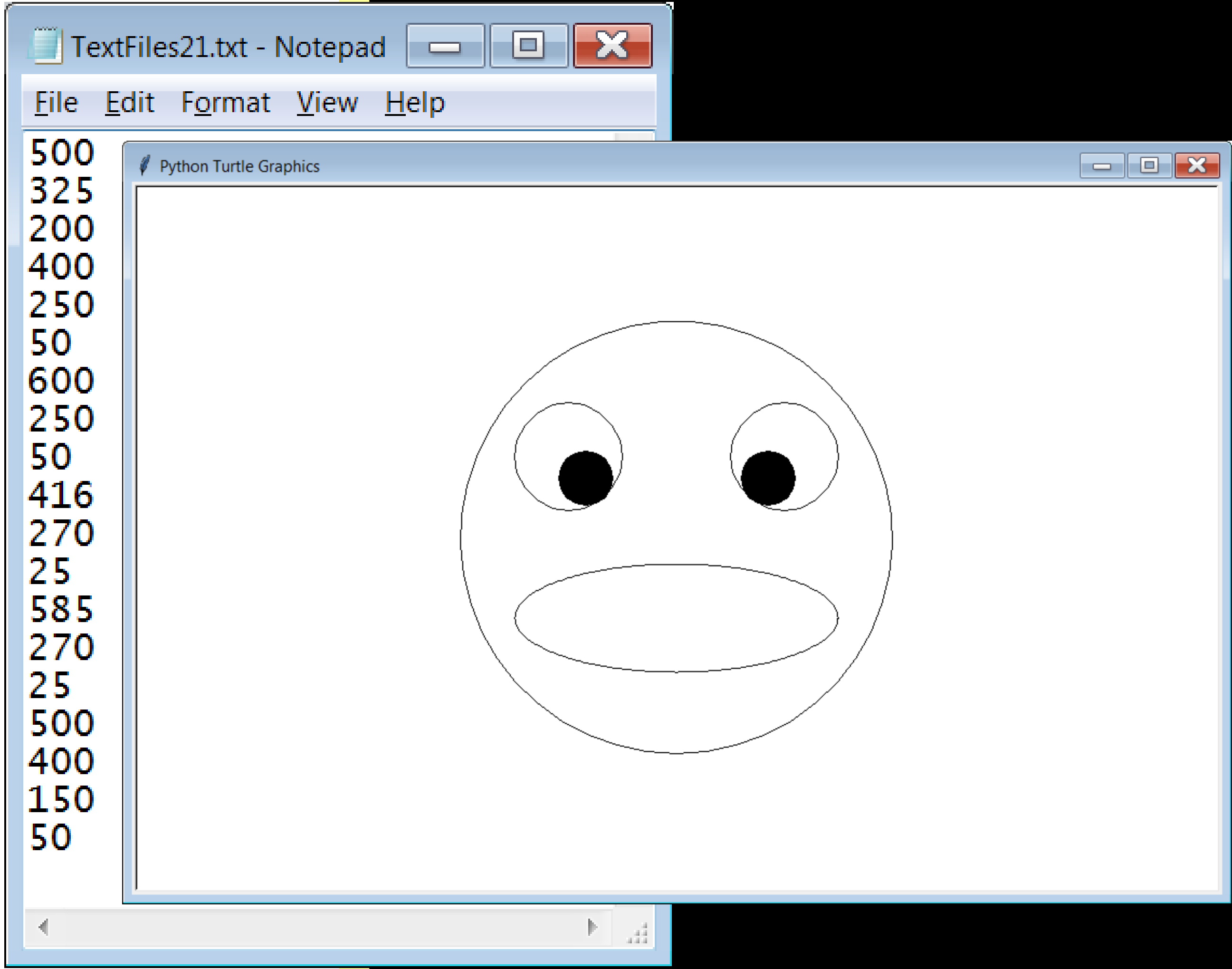
Using Text File Data in Graphics Programs

```
# TextFiles21.py
# This program demonstrates that data from a
# text file can be used in a graphics program.


from Graphics import *


file = open("TextFiles12.txt",'r')

x1 = int(file.readline())
y1 = int(file.readline())
r1 = int(file.readline())
x2 = int(file.readline())
y2 = int(file.readline())
r2 = int(file.readline())
x3 = int(file.readline())
y3 = int(file.readline())
r3 = int(file.readline())
x4 = int(file.readline())
y4 = int(file.readline())
r4 = int(file.readline())
x5 = int(file.readline())
y5 = int(file.readline())
r5 = int(file.readline())
x6 = int(file.readline())
y6 = int(file.readline())
r6 = int(file.readline())
r7 = int(file.readline())

file.close()

beginGrfx(1000,650)

drawCircle(x1,y1,r1)
drawCircle(x2,y2,r2)
drawCircle(x3,y3,r3)
fillCircle(x4,y4,r4)
fillCircle(x5,y5,r5)
drawOval(x6,y6,r6,r7)

endGrfx()
```



```
500
325
200
400
250
50
600
250
50
416
270
25
585
270
25
500
400
150
50
```

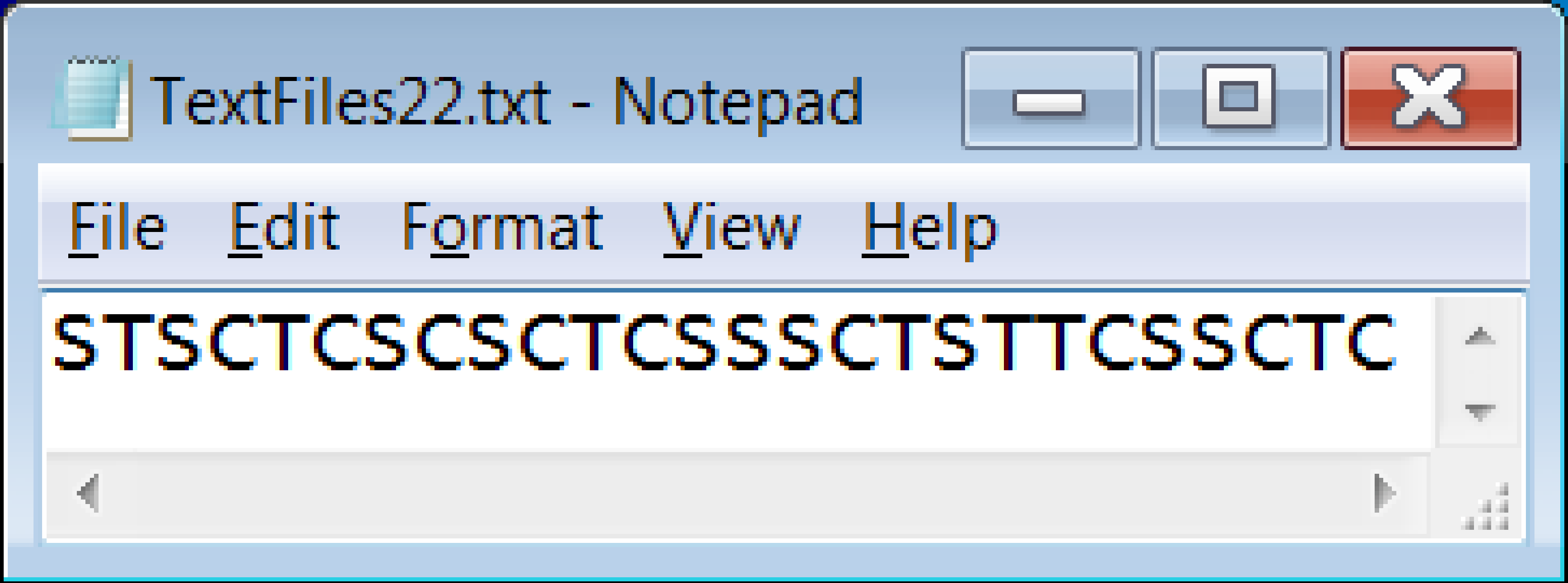
```python
# TextFiles21.py
# This program demonstrates that data from a
# text file can be used in a graphics program.


from Graphics import *


file = open("TextFiles12.txt",'r')

x1 = int(file.readline())
y1 = int(file.readline())
r1 = int(file.readline())
x2 = int(file.readline())
y2 = int(file.readline())
r2 = int(file.readline())
x3 = int(file.readline())
y3 = int(file.readline())
r3 = int(file.readline())
x4 = int(file.readline())
y4 = int(file.readline())
r4 = int(file.readline())
x5 = int(file.readline())
y5 = int(file.readline())
r5 = int(file.readline())
x6 = int(file.readline())
y6 = int(file.readline())
r6 = int(file.readline())
r7 = int(file.readline())

file.close()

beginGrfx(1000,650)

drawCircle(x1,y1,r1)
drawCircle(x2,y2,r2)
drawCircle(x3,y3,r3)
fillCircle(x4,y4,r4)
fillCircle(x5,y5,r5)
drawOval(x6,y6,r6,r7)

endGrfx()
```



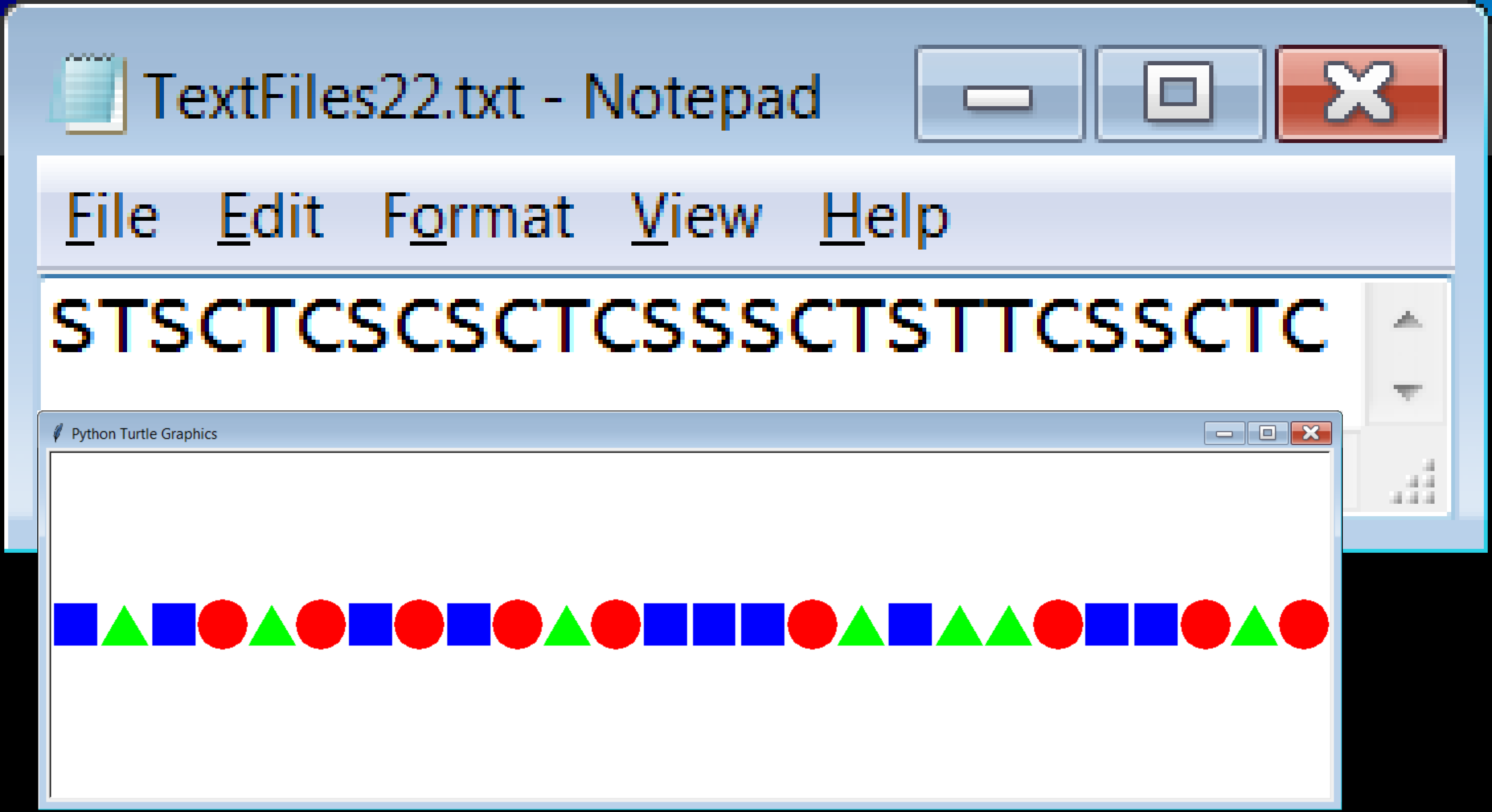
```
500
325
200
400
250
50
600
250
50
416
270
25
585
270
25
500
400
150
50
```

```python
# TextFiles22.py
# This program demonstrates that non-numerical
# data can also be used in a graphics program.


from Graphics import *


def drawBlueSquare(x):
    setColor("blue")
    x = x * 50 + 25
    fillRegularPolygon(x,175,30,4)


def drawRedCircle(x):
    setColor("red")
    x = x * 50 + 25
    fillCircle(x,175,25)


def drawGreenTriangle(x):
    setColor("green")
    x = x * 50 + 25
    fillRegularPolygon(x,183,27,3)
```

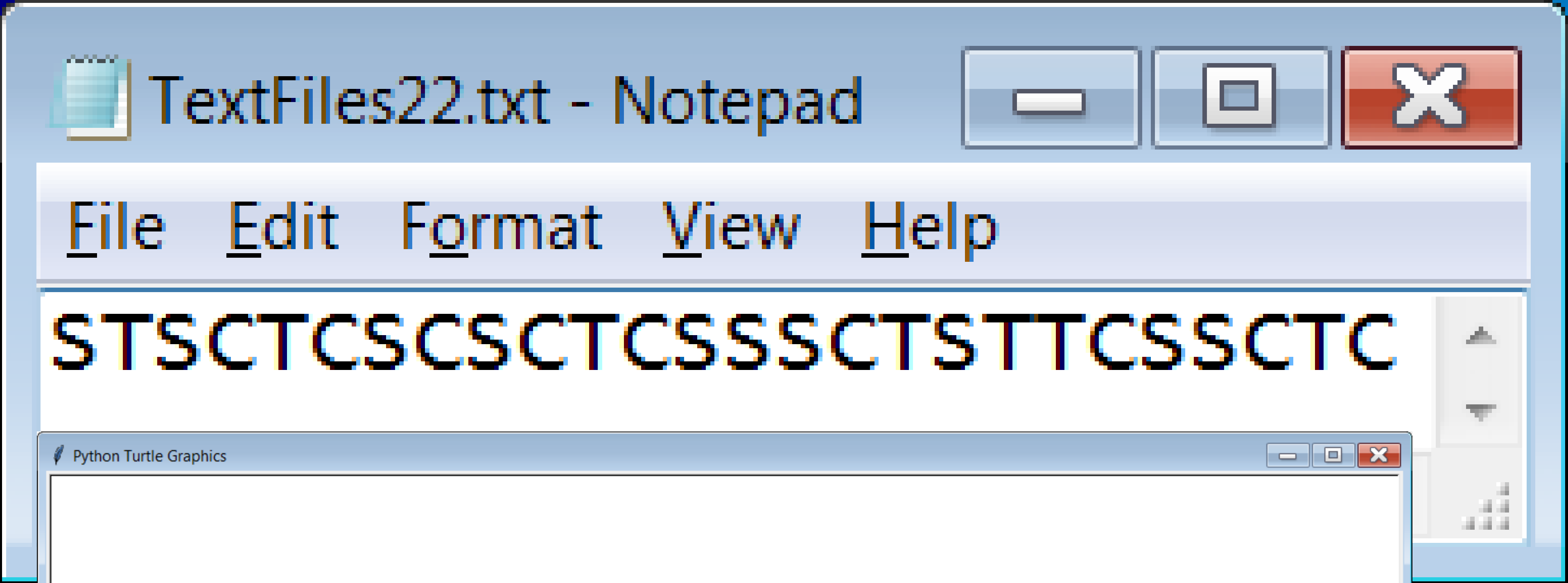
```python



###########
#   MAIN   #
###########

file = open("TextFiles22.txt",'r')
lineOfText = file.readline()
file.close()

beginGrfx(1300,350)

for x in range(len(lineOfText)):
   if lineOfText[x] == "S":
      drawBlueSquare(x)
   elif lineOfText[x] == "C":
      drawRedCircle(x)
   elif lineOfText[x] == "T":
      drawGreenTriangle(x)

endGrfx()


```



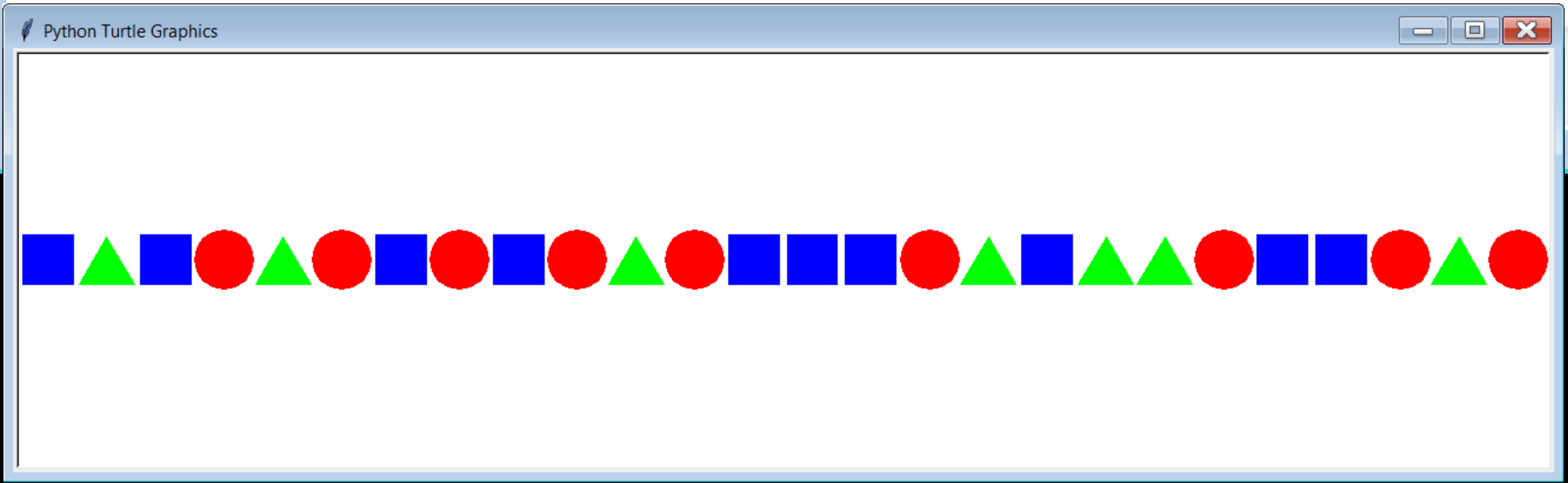
```
STSCTCSCSCTCSSSCTSTTCSSCTC
```



```
STSCTCSCSCTCSSSCTSTTCSSCTC
```

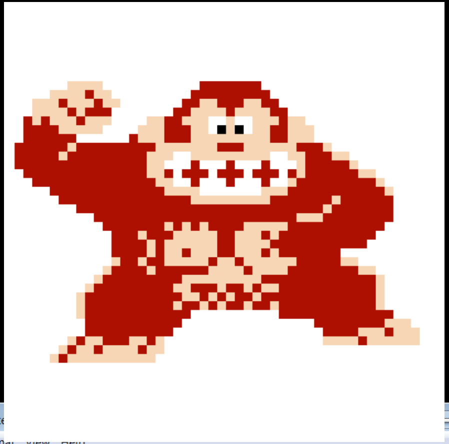


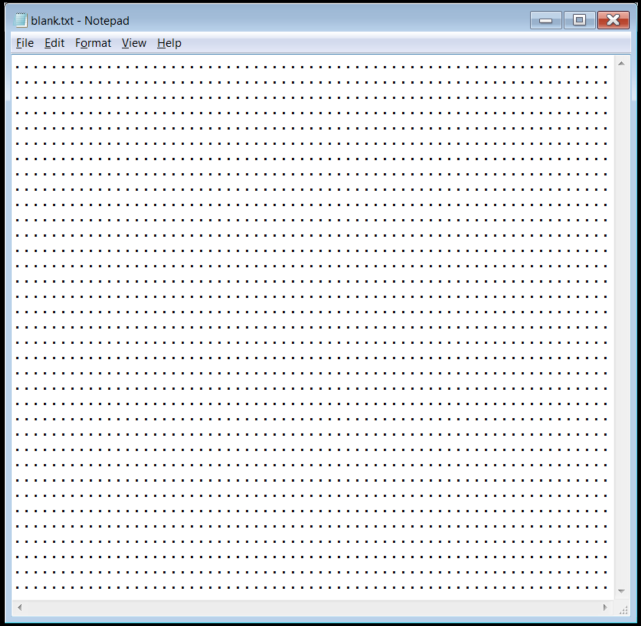
```
STSCTCSCSCTCSSSCTSTTCSSCTC
```

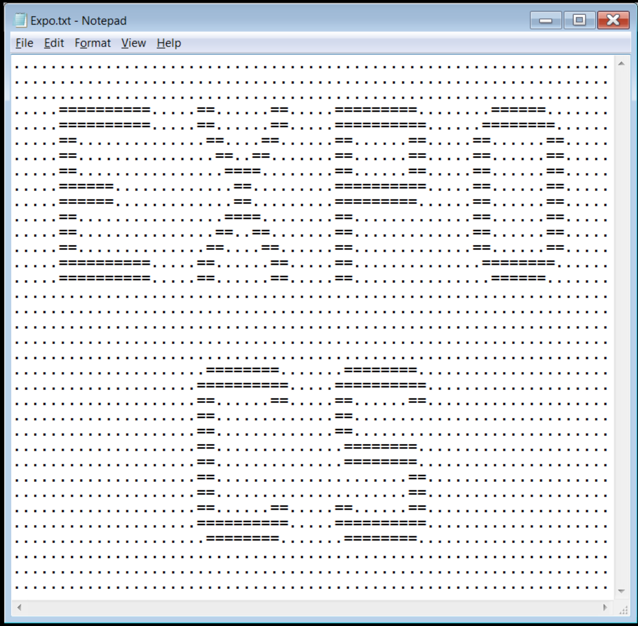


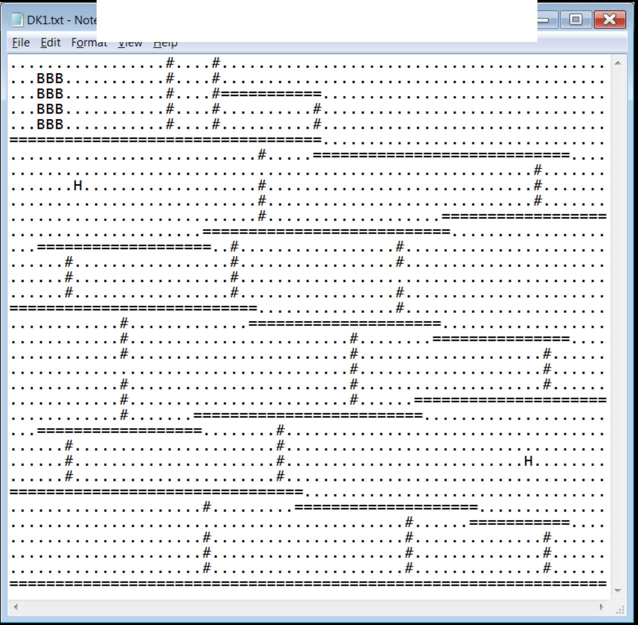
Note how the graphics output precisely correlates to the characters in the text file. Try altering the characters in this file and run the program again to see what happens.

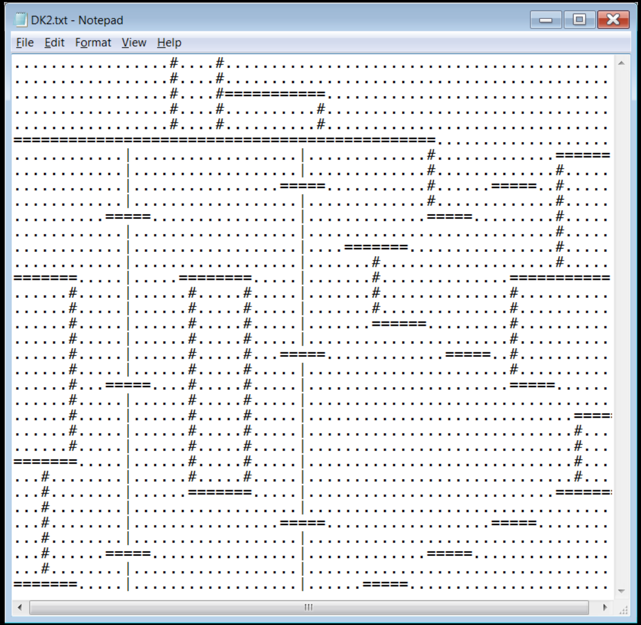
TextFiles23.py Data Files

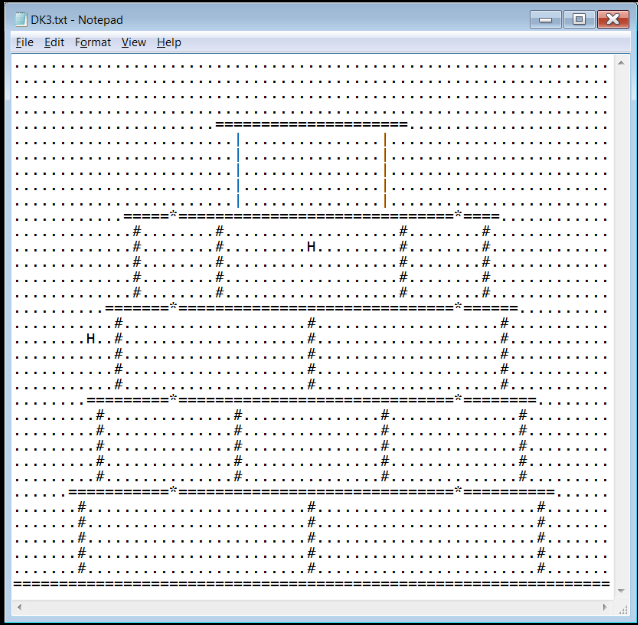








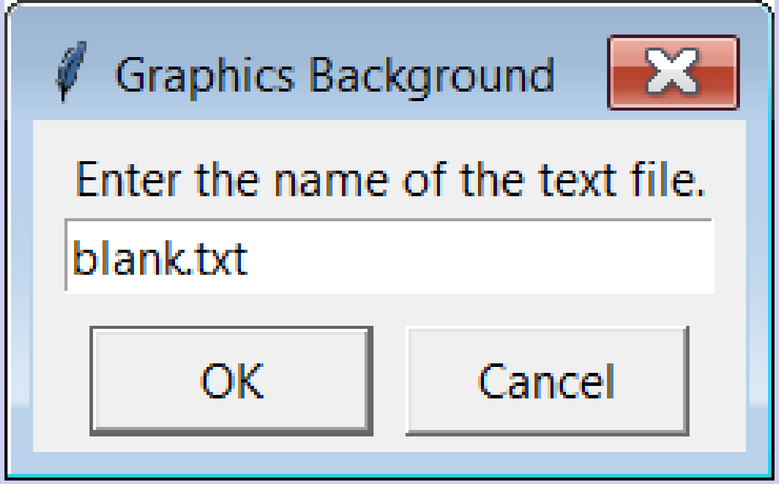




```python
# TextFiles23.py
# This program will ask the user to enter
# the name of a text file, and then use that
# text file to display a graphics background.
# Students are provided with 5 example files:
# blank.txt, Expo.txt, DK1.txt, DK2.txt, and
# DK3.txt -- the latter 3 resemble stages from
# Nintendo's Donkey Kong arcade game.
# NOTE: This same program is used for Lab 16B.


from Graphics import *
import os


def convert(q):
   return q * 20


def drawSpace(r,c):
   x = convert(c)
   y = convert(r)
   setColor("black")
   fillRect(x,y,20,20)


def drawGirder(r,c):
   x = convert(c)
   y = convert(r)
   setColor("red");
   fillRect(x,y,20,20)
   setColor("black")
   fillCircle(x+10,y+9,6)


def drawLadder(r,c):
   x = convert(c)
   y = convert(r)
   setColor("black")
   fillRect(x,y,20,20)
   setColor("white")
   fillRect(x,y,3,20)
   fillRect(x+16,y,4,20)
   fillRect(x,y+8,20,4)


def drawHammer(r,c):
   x = convert(c)
   y = convert(r)
   setColor("black")
   fillRect(x,y,20,20)
   setColor(150,100,15)
   fillRect(x,y,20,10)
   setColor("yellow");
   fillRect(x+8,y+10,4,10)


def drawBarrel(r,c):
   x = convert(c)
   y = convert(r)
   setColor("black")
   fillRect(x,y,20,20)
   setColor(150,100,15)
   fillRect(x+5,y,10,20)
   fillArc(x+5,y+10,5,10,180,360)
   fillArc(x+15,y+10,5,10,0,180)
   setColor("black")
   drawLine(x+16,y+9,x+16,y+13)
   drawLine(x+15,y+6,x+15,y+16)
   setColor("white")
   drawLine(x+5,y+4,x+5,y+14)
   drawLine(x+4,y+7,x+4,y+10)
   setColor(211,211,211)
   drawLine(x+5,y,x+15,y)
```

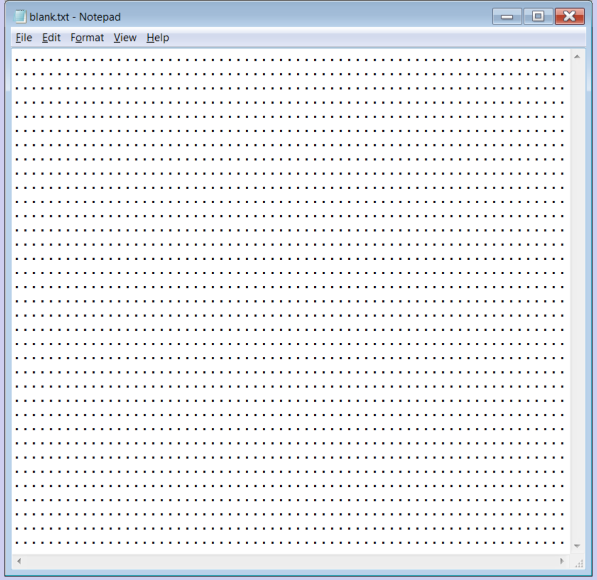
```python


def drawLock(r,c):
   x = convert(c)
   y = convert(r)
   setColor("cyan")
   fillRect(x,y,20,5)
   setColor("yellow")
   fillRect(x,y+5,20,15)


def drawPole(r,c):
   x = convert(c)
   y = convert(r)
   setColor("black")
   fillRect(x,y,20,20)
   setColor("cyan")
   fillRect(x+7,y,6,20)


def drawUnknown(r,c):
   x = convert(c)
   y = convert(r)
   setColor("pink")
   fillRect(x,y,20,20)
   setColor("black")
   drawString("?",x+3,y+25,"Courier",16,"bold")



```

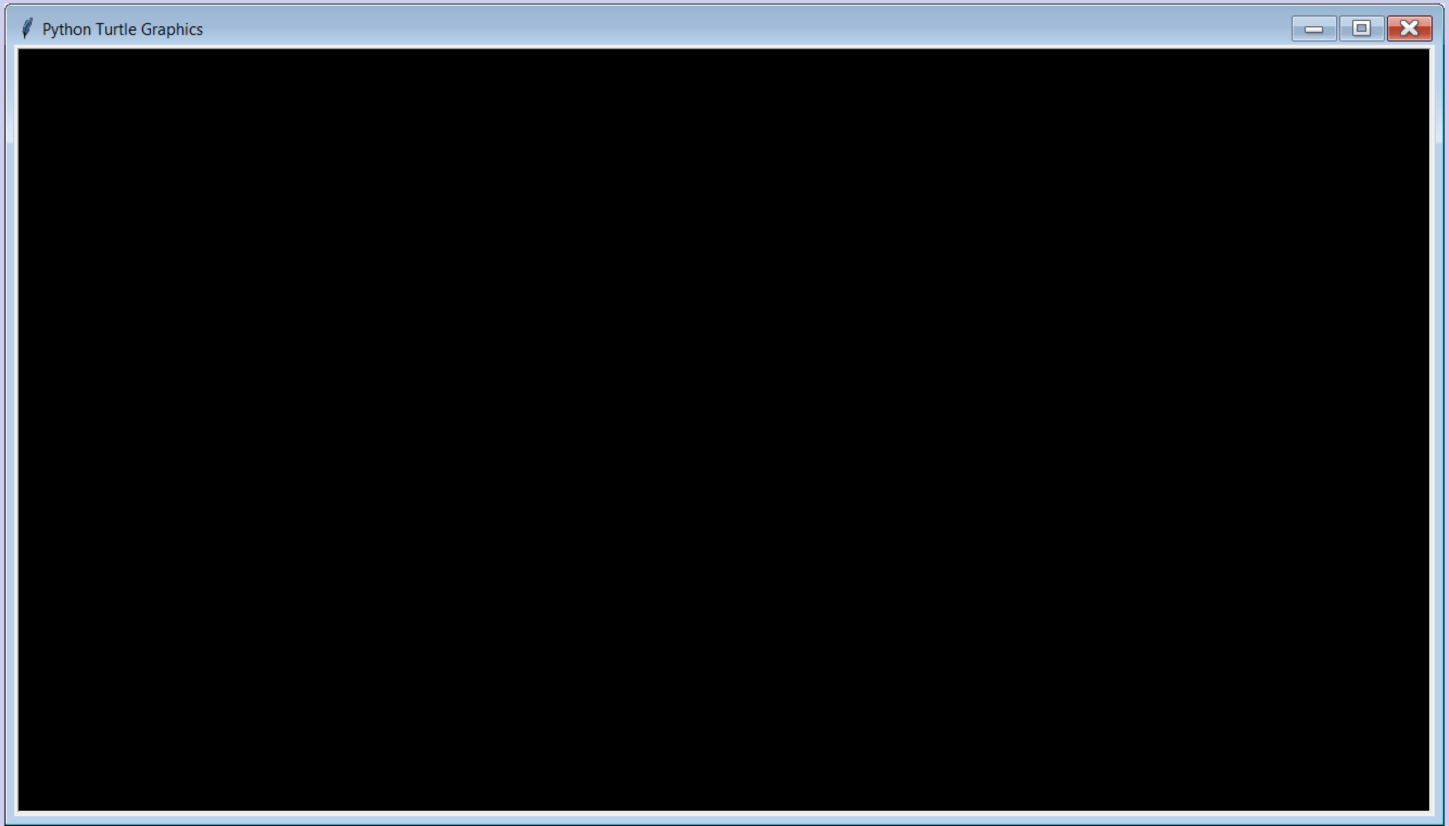
```python
##########
#  MAIN  #
##########

numRows = 35
numCols = 65
fileName = textinput("Graphics Background","Enter the name of the text file.")
while not os.path.exists(fileName):
   fileName = textinput("File does not exist.","Enter the name of the text file.")
file = open(fileName,"r")
background = file.readlines()
file.close()

beginGrfx(1300,700)

for r in range(numRows):
   for c in range(numCols):
      if background[r][c] == '.':
         drawSpace(r,c)
      elif background[r][c] == '=':
         drawGirder(r,c)
      elif background[r][c] == '#':
         drawLadder(r,c)
      elif background[r][c] == 'H':
         drawHammer(r,c)
      elif background[r][c] == 'B':
         drawBarrel(r,c)
      elif background[r][c] == '*':
         drawLock(r,c)
      elif background[r][c] == '|':
         drawPole(r,c)
      else:
         drawUnknown(r,c)
   update()

endGrfx()
```

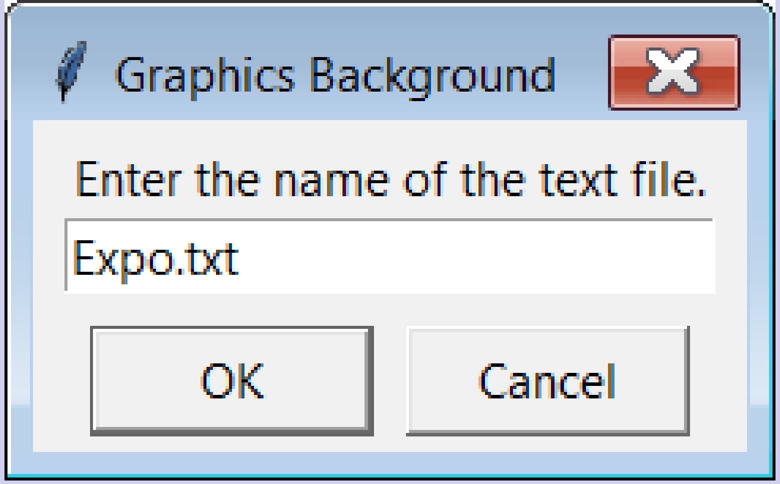


Enter the name of the text file.

blank.txt

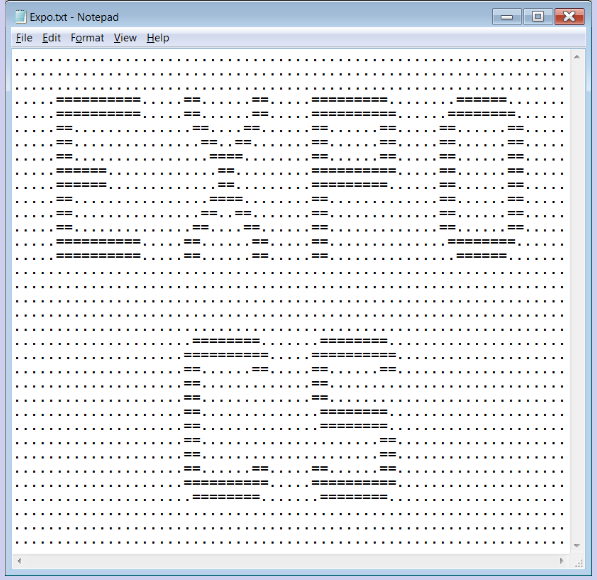




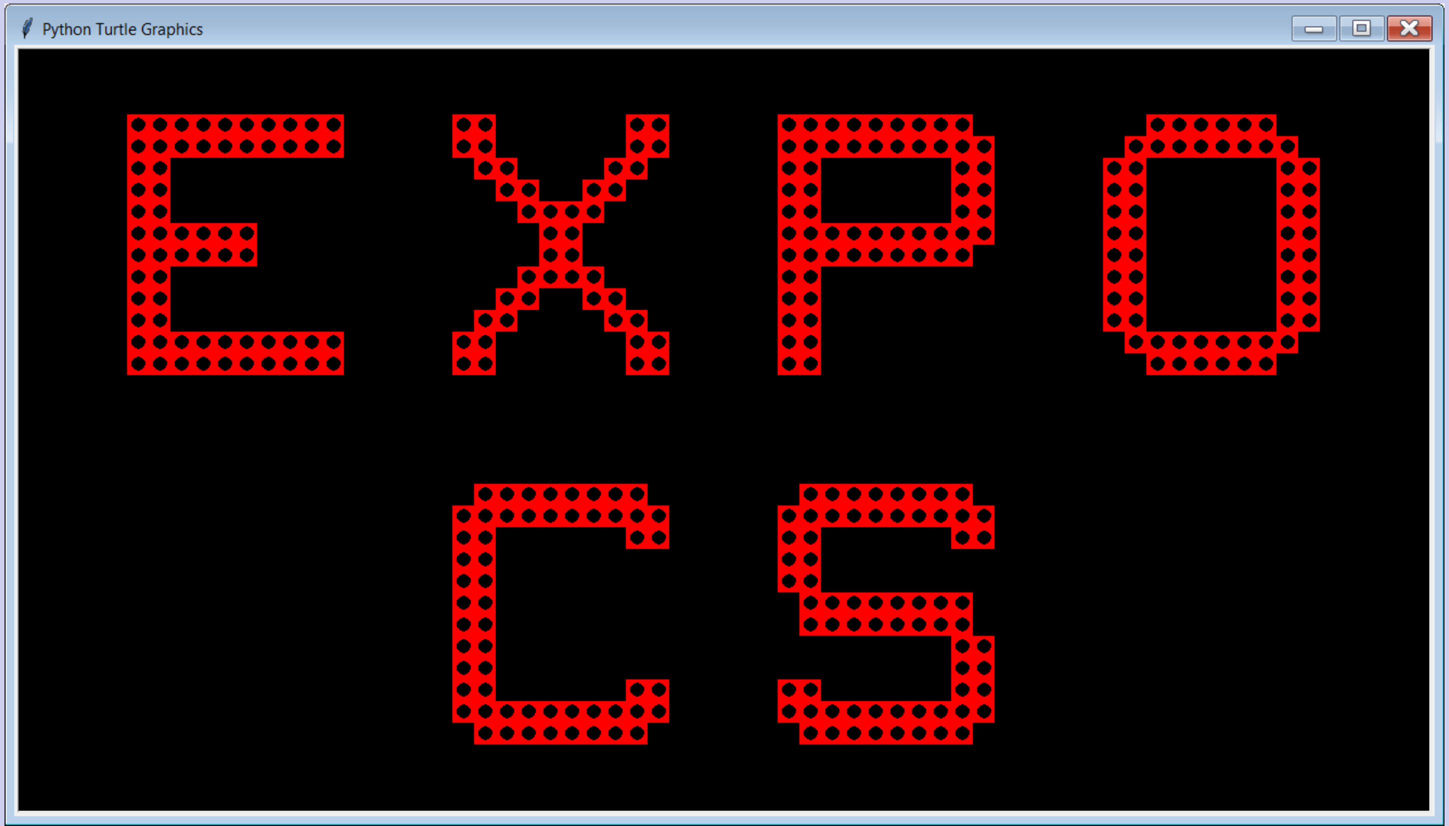


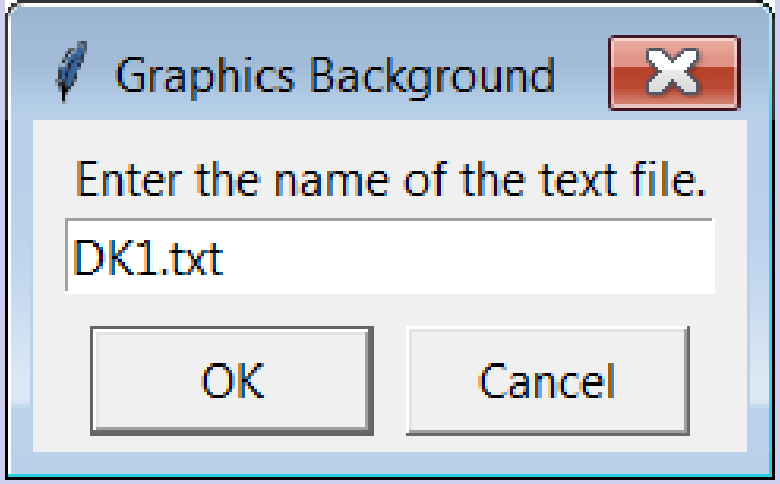
Enter the name of the text file.

Expo.txt



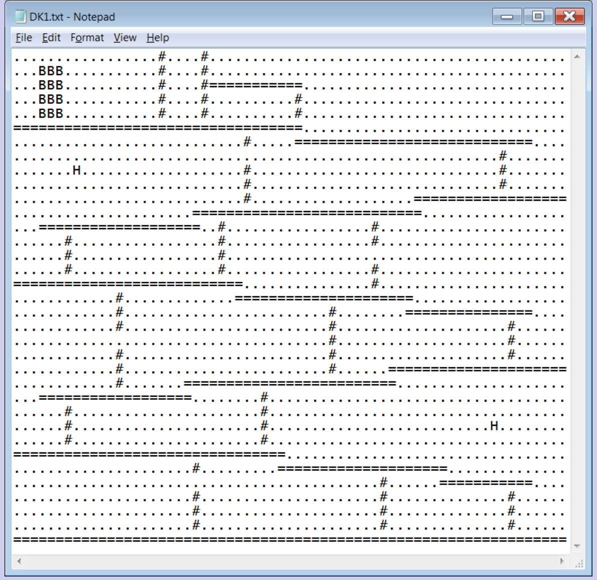
```
.................................................................
.................................................................
.................................................................
.....==========.....==......==.....=========.......======.......
.....==========.....==......==.....==========......========......
.....==..............==....==......==......==.....==......==.....
.....==...............==..==.......==......==.....==......==.....
.....==................====........==......==.....==......==.....
.....======.............==.........==========.....==......==.....
.....======.............==.........=========......==......==.....
.....==................====........==.............==......==.....
.....==...............==..==.......==.............==......==.....
.....==..............==....==......==.............==......==.....
.....==========.....==......==.....==..............========......
.....==========.....==......==.....==...............======.......
.................................................................
.................................................................
.................................................................
.................................................................
.................................................................
.....................========.......========.....................
....................==========.....==========....................
....................==......==.....==......==....................
....................==.............==............................
....................==.............==............................
....................==..............========.....................
....................==..............========.....................
....................==.....................==....................
....................==.....................==....................
....................==......==.....==......==....................
....................==========.....==========....................
.....................========.......========.....................
.................................................................
.................................................................
.................................................................
```



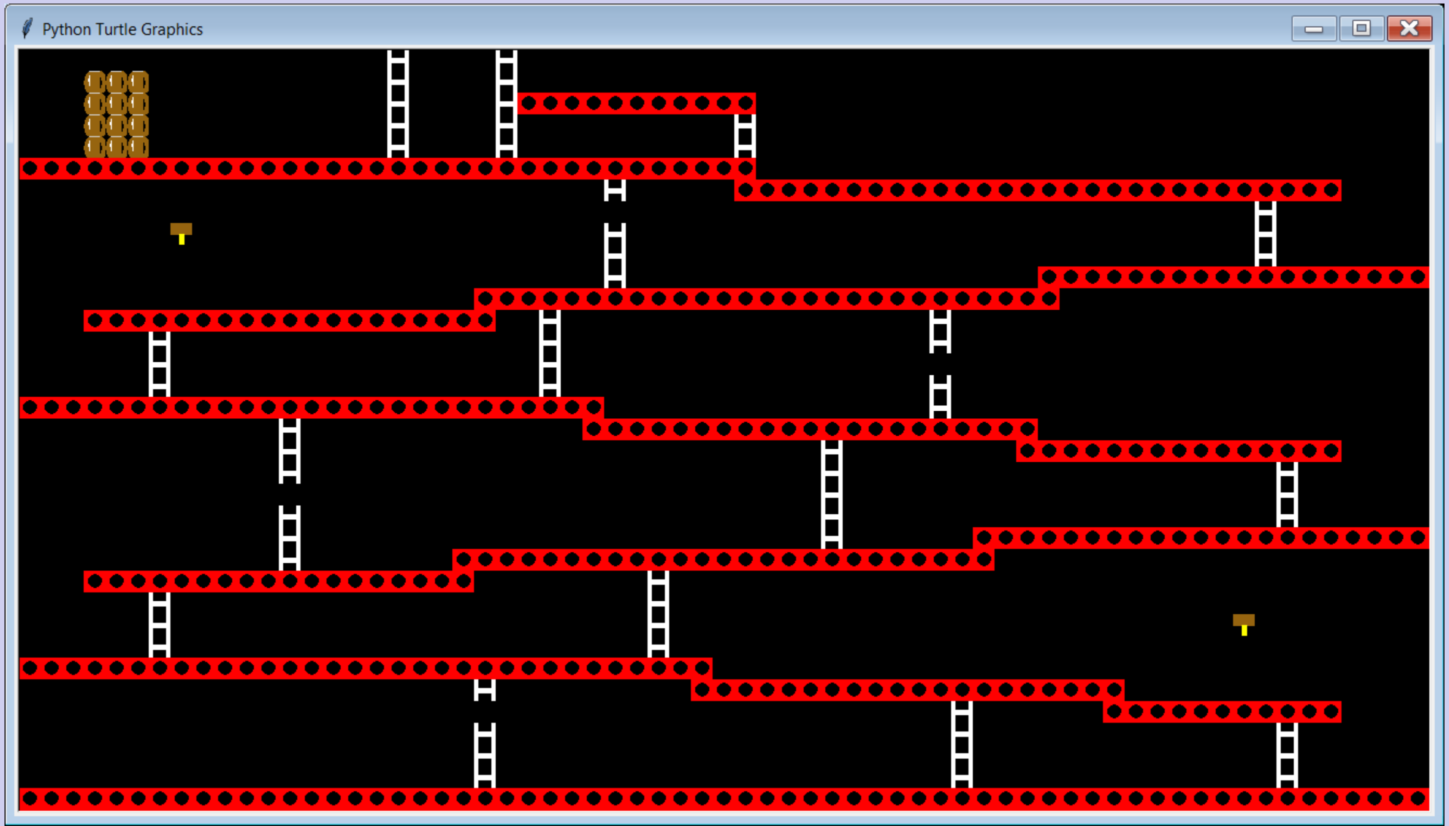


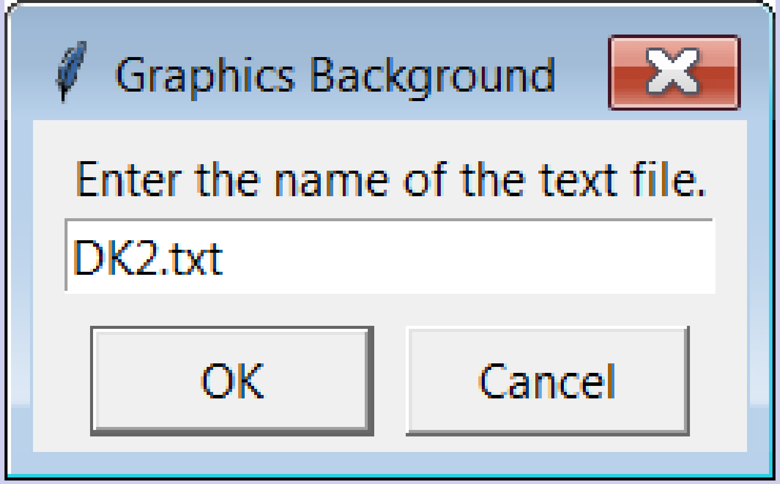
Enter the name of the text file.

DK1.txt



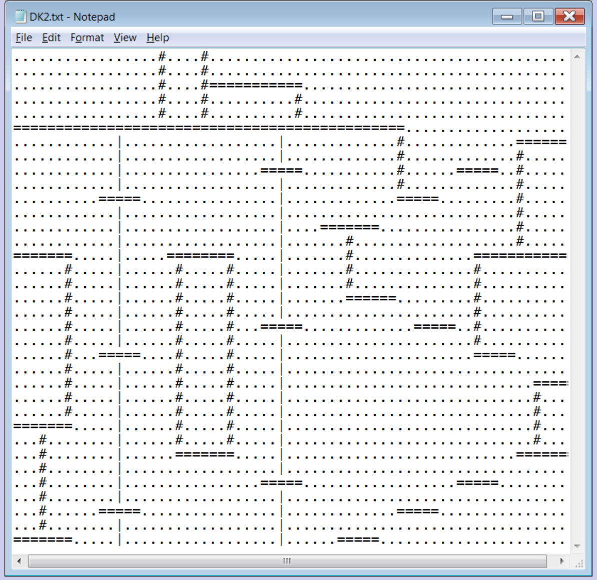
```
.................#....#..........................................
...BBB...........#....#..........................................
...BBB...........#....#===========...............................
...BBB...........#....#..........#...............................
...BBB...........#....#..........#...............................
==================================...............................
...........................#.....============================....
.........................................................#.......
.......H...................#.............................#.......
...........................#.............................#.......
...........................#...................==================
.....................===========================.................
...===================..#.................#......................
......#.................#.................#......................
......#.................#........................................
......#.................#.................#......................
===========================...............#......................
............#.............=====================..................
............#........................#........===============....
............#........................#....................#......
.....................................#....................#......
............#........................#....................#......
............#........................#......=====================
............#.......=========================....................
...==================........#...................................
......#......................#...................................
......#......................#..........................H........
......#......................#...................................
================================.................................
.....................#.........====================..............
...........................................#......===========....
.....................#.....................#..............#......
.....................#.....................#..............#......
.....................#.....................#..............#......
=================================================================
```



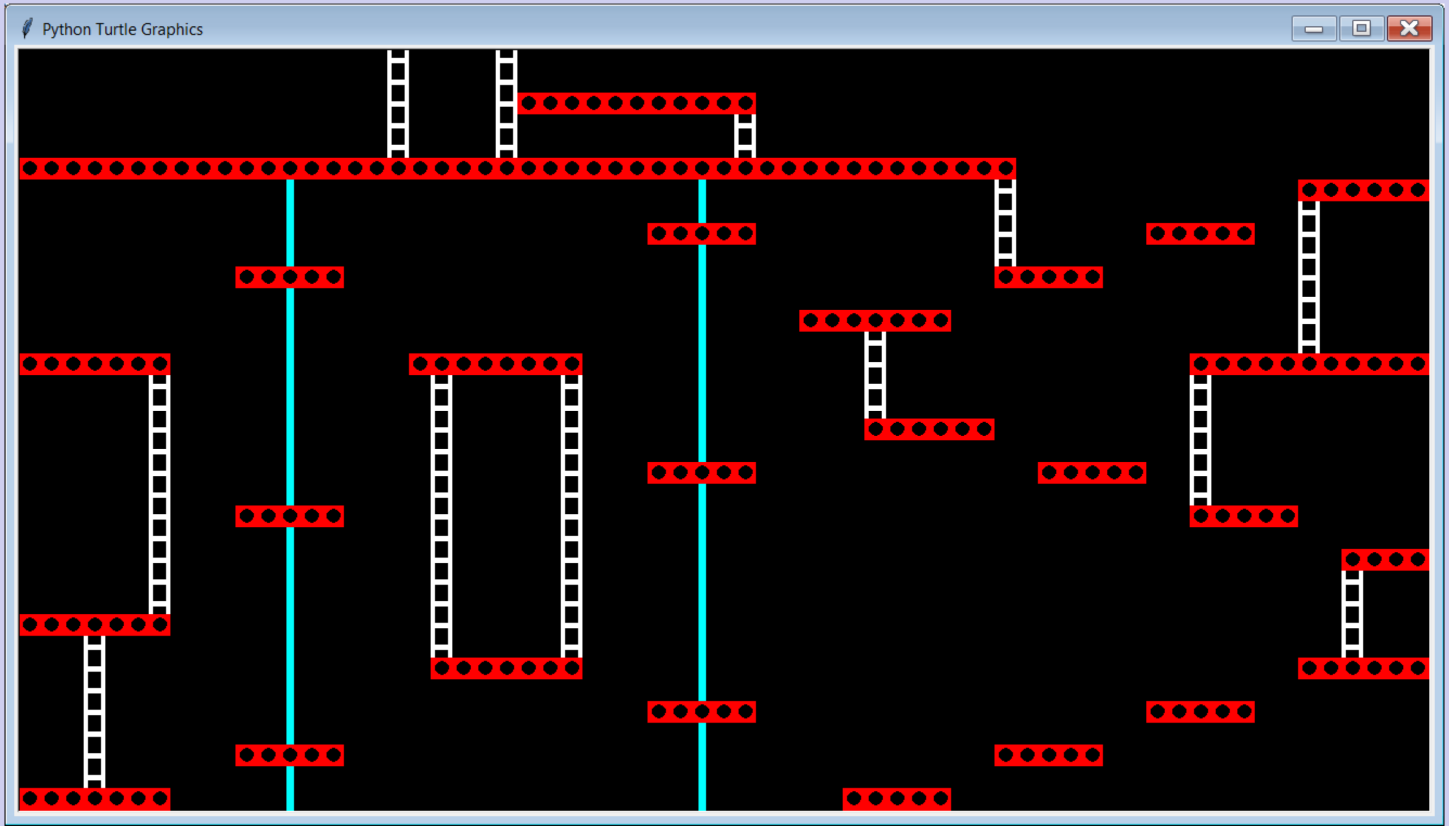

Graphics Background

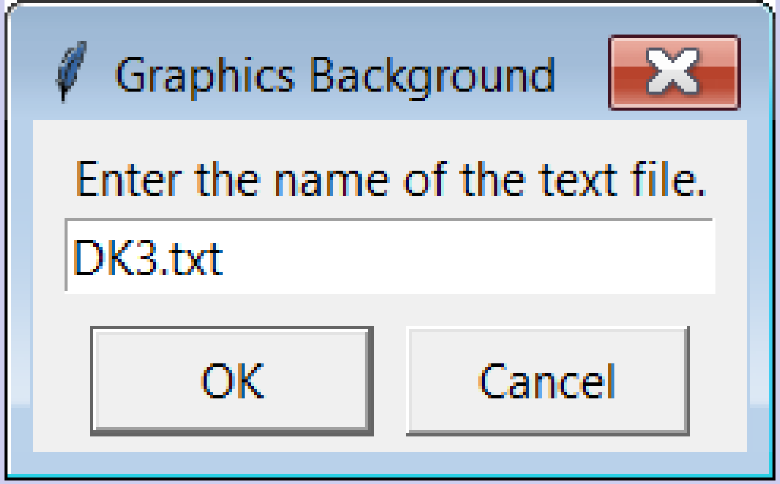
Enter the name of the text file.

DK2.txt


DK2.txt - Notepad

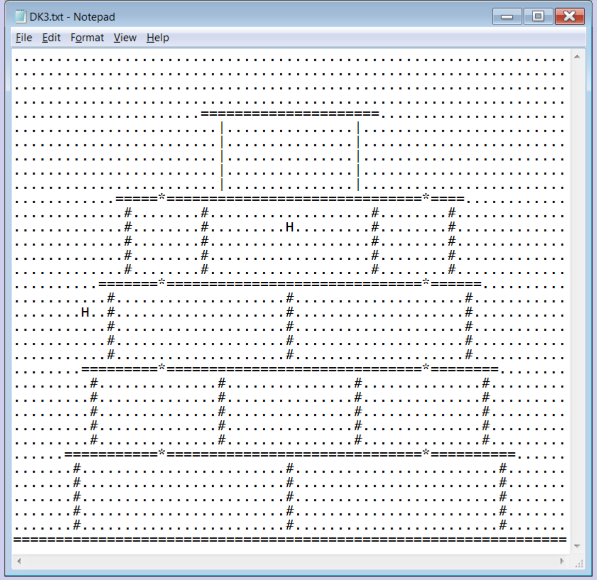
```
.................#....#..........................................
.................#....#..........................................
.................#....#===========...............................
.................#....#..........#...............................
.................#....#..........#...............................
==============================================...................
............|..................|.............#.............======
............|..................|.............#.............#.....
............|................=====...........#......=====..#.....
............|..................|.............#.............#.....
..........=====................|.............=====.........#.....
............|..................|...........................#.....
............|..................|....=======................#.....
............|..................|.......#...................#.....
=======.....|.....========.....|.......#..............===========
......#.....|......#.....#.....|.......#..............#..........
......#.....|......#.....#.....|.......#..............#..........
......#.....|......#.....#.....|.......======.........#..........
......#.....|......#.....#.....|......................#..........
......#.....|......#.....#...=====.............=====..#..........
......#.....|......#.....#.....|......................#..........
......#...=====....#.....#.....|......................=====......
......#.....|......#.....#.....|.................................
......#.....|......#.....#.....|.............................====
......#.....|......#.....#.....|.............................#...
......#.....|......#.....#.....|.............................#...
=======.....|......#.....#.....|.............................#...
...#........|......#.....#.....|.............................#...
...#........|......=======.....|...........................======
...#........|..................|.................................
...#........|................=====..................=====........
...#........|..................|.................................
...#......=====................|.............=====...............
...#........|..................|.................................
=======.....|..................|......=====......................
```



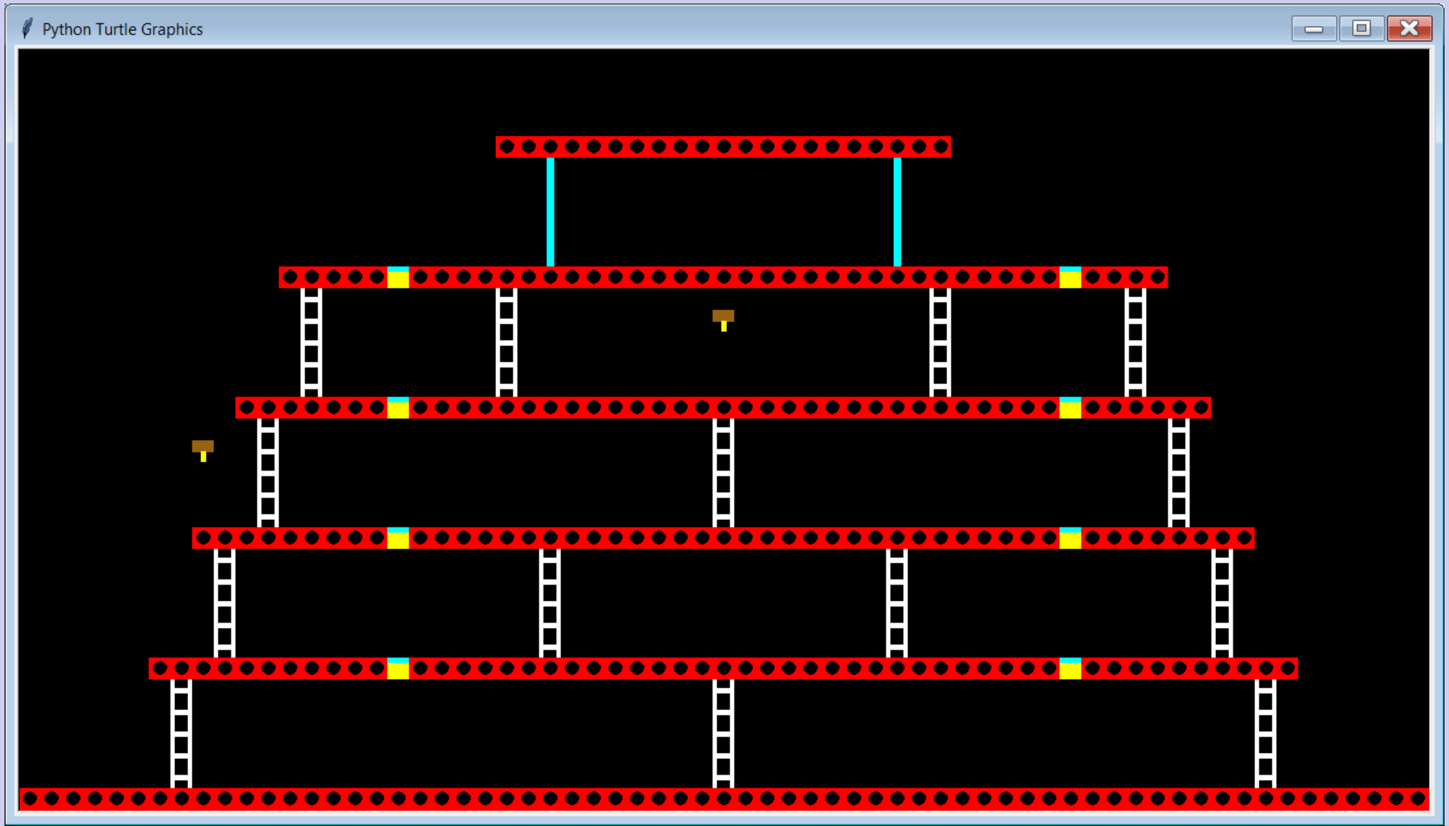


Enter the name of the text file.

DK3.txt



```
.................................................................
.................................................................
.................................................................
.................................................................
......................=====================......................
........................|...............|........................
........................|...............|........................
........................|...............|........................
........................|...............|........................
........................|...............|........................
............=====*==============================*====............
.............#........#...................#........#.............
.............#........#.........H.........#........#.............
.............#........#...................#........#.............
.............#........#...................#........#.............
.............#........#...................#........#.............
..........=======*==============================*======..........
...........#....................#....................#...........
........H..#....................#....................#...........
...........#....................#....................#...........
...........#....................#....................#...........
...........#....................#....................#...........
........=========*==============================*========........
.........#..............#...............#..............#.........
.........#..............#...............#..............#.........
.........#..............#...............#..............#.........
.........#..............#...............#..............#.........
.........#..............#...............#..............#.........
......===========*==============================*==========......
.......#........................#........................#.......
.......#........................#........................#.......
.......#........................#........................#.......
.......#........................#........................#.......
.......#........................#........................#.......
=================================================================
```


Python Turtle Graphics