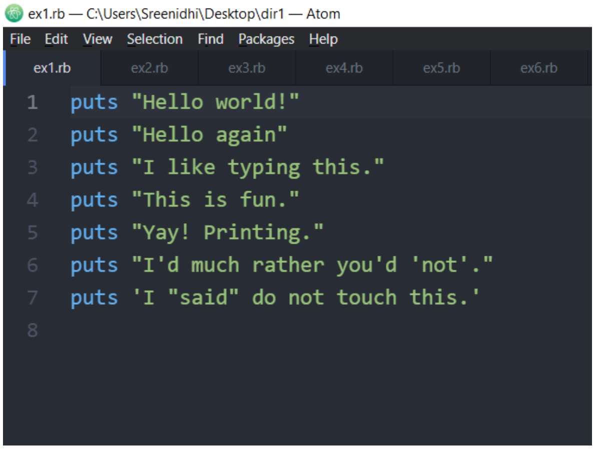
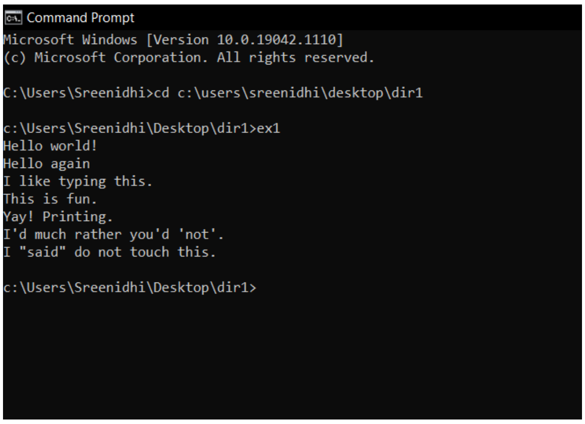
Exercise 1

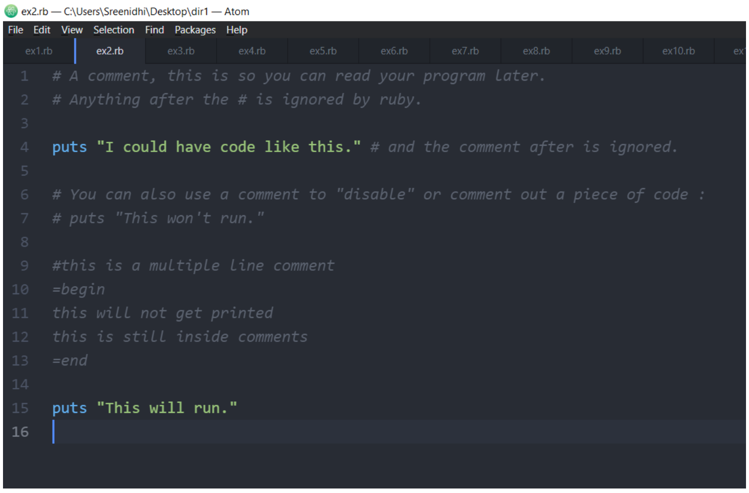

ex1.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb
ex2.rb
ex3.rb
ex4.rb
ex5.rb
ex6.rb

```ruby
puts "Hello world!"
puts "Hello again"
puts "I like typing this."
puts "This is fun."
puts "Yay! Printing."
puts "I'd much rather you'd 'not'."
puts 'I "said" do not touch this.'

```

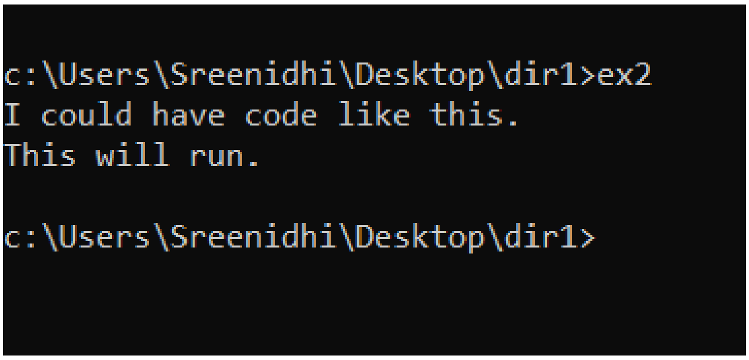


```
Microsoft Windows [Version 10.0.19042.1110]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sreenidhi>cd c:\users\sreenidhi\desktop\dir1

c:\Users\Sreenidhi\Desktop\dir1>ex1
Hello world!
Hello again
I like typing this.
This is fun.
Yay! Printing.
I'd much rather you'd 'not'.
I "said" do not touch this.

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 2

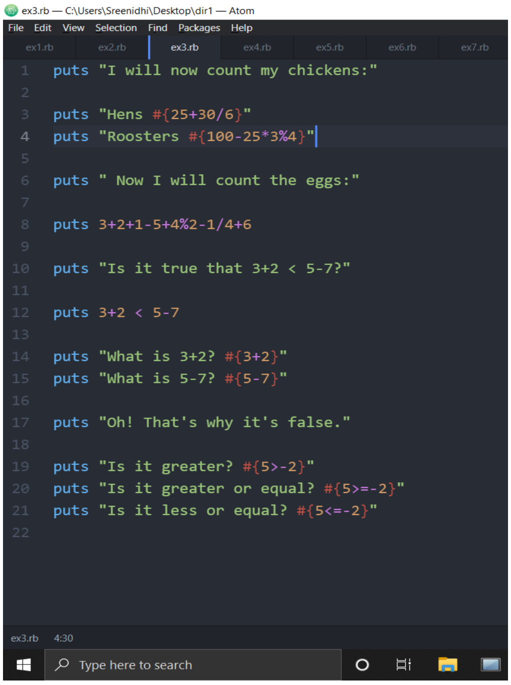

ex2.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb
ex2.rb
ex3.rb
ex4.rb
ex5.rb
ex6.rb
ex7.rb
ex8.rb
ex9.rb
ex10.rb
ex

```ruby
# A comment, this is so you can read your program later.
# Anything after the # is ignored by ruby.

puts "I could have code like this." # and the comment after is ignored.

# You can also use a comment to "disable" or comment out a piece of code :
# puts "This won't run."

#this is a multiple line comment
=begin
this will not get printed
this is still inside comments
=end

puts "This will run."

```

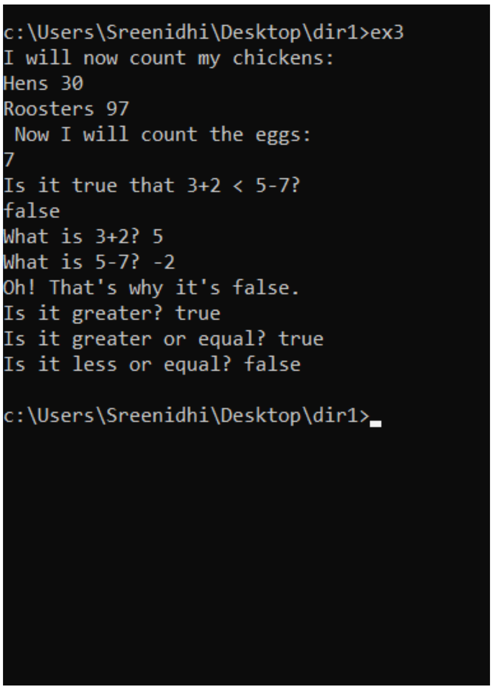


```
c:\Users\Sreenidhi\Desktop\dir1>ex2
I could have code like this.
This will run.

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 3

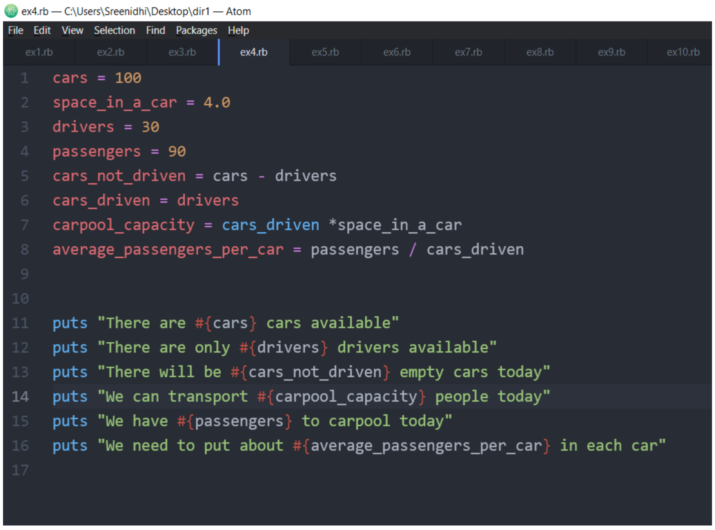

ex3.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb
ex3.rb 4:30
Type here to search

```ruby
puts "I will now count my chickens:"

puts "Hens #{25+30/6}"
puts "Roosters #{100-25*3%4}"

puts " Now I will count the eggs:"

puts 3+2+1-5+4%2-1/4+6

puts "Is it true that 3+2 < 5-7?"

puts 3+2 < 5-7

puts "What is 3+2? #{3+2}"
puts "What is 5-7? #{5-7}"

puts "Oh! That's why it's false."

puts "Is it greater? #{5>-2}"
puts "Is it greater or equal? #{5>=-2}"
puts "Is it less or equal? #{5<=-2}"
```

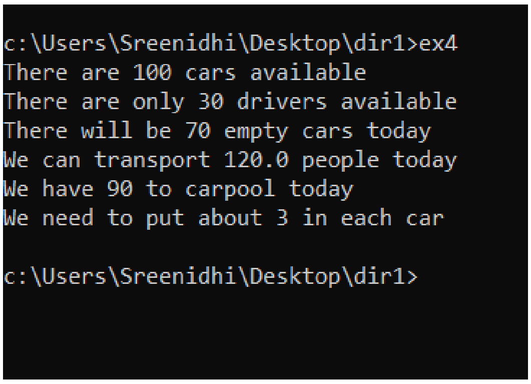


```
c:\Users\Sreenidhi\Desktop\dir1>ex3
I will now count my chickens:
Hens 30
Roosters 97
 Now I will count the eggs:
7
Is it true that 3+2 < 5-7?
false
What is 3+2? 5
What is 5-7? -2
Oh! That's why it's false.
Is it greater? true
Is it greater or equal? true
Is it less or equal? false

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 4

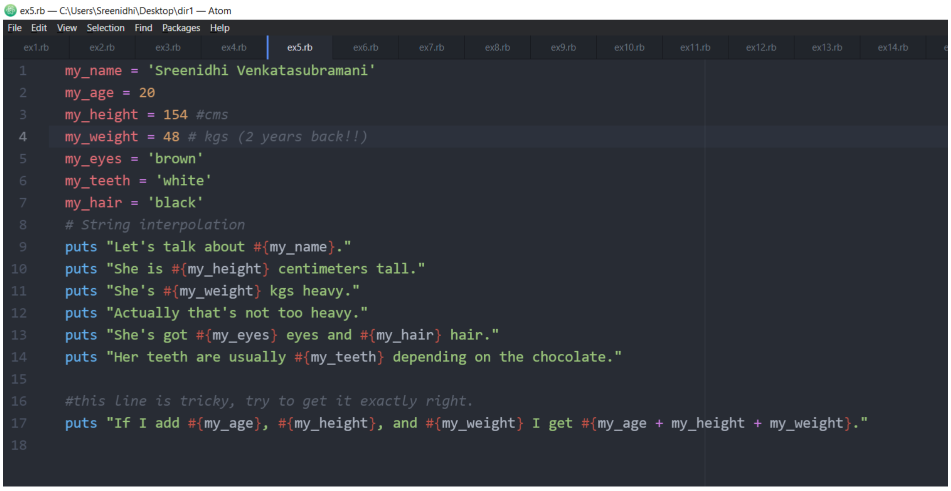

ex4.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb ex8.rb ex9.rb ex10.rb

```ruby
cars = 100
space_in_a_car = 4.0
drivers = 30
passengers = 90
cars_not_driven = cars - drivers
cars_driven = drivers
carpool_capacity = cars_driven *space_in_a_car
average_passengers_per_car = passengers / cars_driven


puts "There are #{cars} cars available"
puts "There are only #{drivers} drivers available"
puts "There will be #{cars_not_driven} empty cars today"
puts "We can transport #{carpool_capacity} people today"
puts "We have #{passengers} to carpool today"
puts "We need to put about #{average_passengers_per_car} in each car"
```

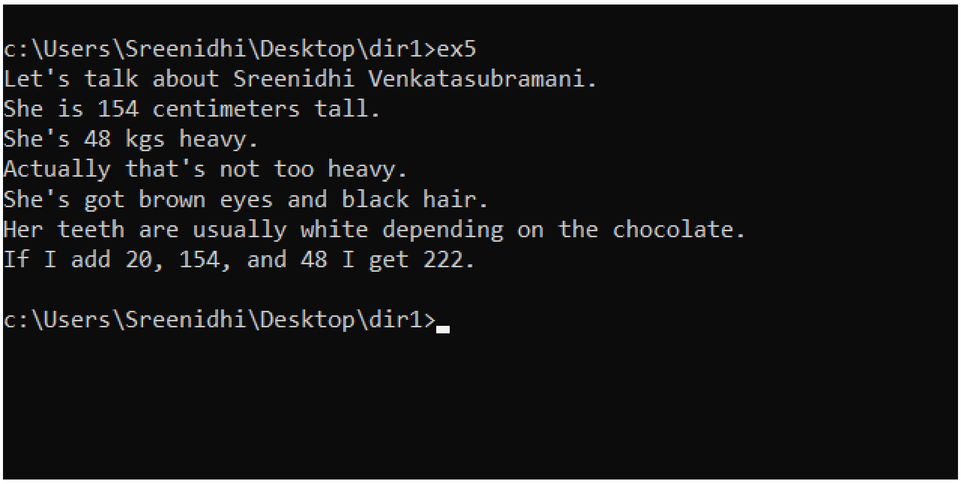


```
c:\Users\Sreenidhi\Desktop\dir1>ex4
There are 100 cars available
There are only 30 drivers available
There will be 70 empty cars today
We can transport 120.0 people today
We have 90 to carpool today
We need to put about 3 in each car

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 5

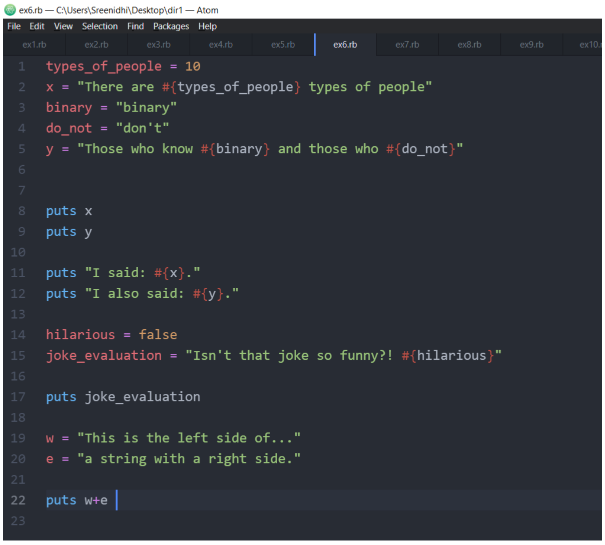

ex5.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb ex8.rb ex9.rb ex10.rb ex11.rb ex12.rb ex13.rb ex14.rb

```ruby
my_name = 'Sreenidhi Venkatasubramani'
my_age = 20
my_height = 154 #cms
my_weight = 48 # kgs (2 years back!!)
my_eyes = 'brown'
my_teeth = 'white'
my_hair = 'black'
# String interpolation
puts "Let's talk about #{my_name}."
puts "She is #{my_height} centimeters tall."
puts "She's #{my_weight} kgs heavy."
puts "Actually that's not too heavy."
puts "She's got #{my_eyes} eyes and #{my_hair} hair."
puts "Her teeth are usually #{my_teeth} depending on the chocolate."

#this line is tricky, try to get it exactly right.
puts "If I add #{my_age}, #{my_height}, and #{my_weight} I get #{my_age + my_height + my_weight}."
```

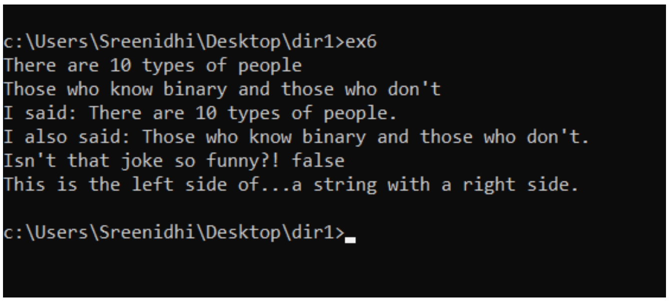


```
c:\Users\Sreenidhi\Desktop\dir1>ex5
Let's talk about Sreenidhi Venkatasubramani.
She is 154 centimeters tall.
She's 48 kgs heavy.
Actually that's not too heavy.
She's got brown eyes and black hair.
Her teeth are usually white depending on the chocolate.
If I add 20, 154, and 48 I get 222.

c:\Users\Sreenidhi\Desktop\dir1>_
```

Exercise 6

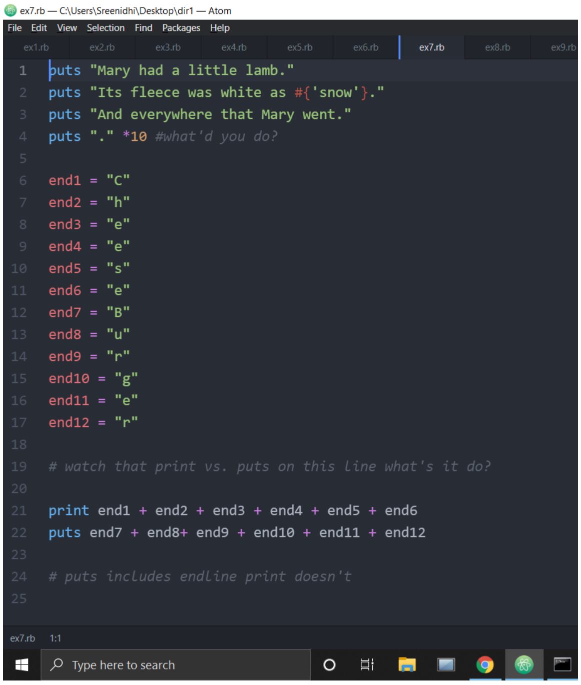

ex6.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb ex8.rb ex9.rb ex10.

```ruby
types_of_people = 10
x = "There are #{types_of_people} types of people"
binary = "binary"
do_not = "don't"
y = "Those who know #{binary} and those who #{do_not}"


puts x
puts y

puts "I said: #{x}."
puts "I also said: #{y}."

hilarious = false
joke_evaluation = "Isn't that joke so funny?! #{hilarious}"

puts joke_evaluation

w = "This is the left side of..."
e = "a string with a right side."

puts w+e
```

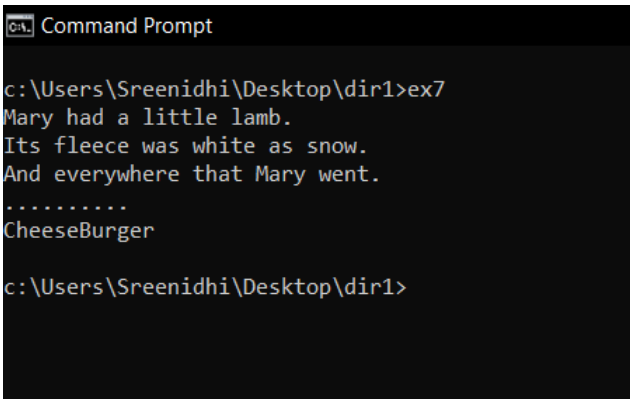


```
c:\Users\Sreenidhi\Desktop\dir1>ex6
There are 10 types of people
Those who know binary and those who don't
I said: There are 10 types of people.
I also said: Those who know binary and those who don't.
Isn't that joke so funny?! false
This is the left side of...a string with a right side.

c:\Users\Sreenidhi\Desktop\dir1>_
```

Exercise 7

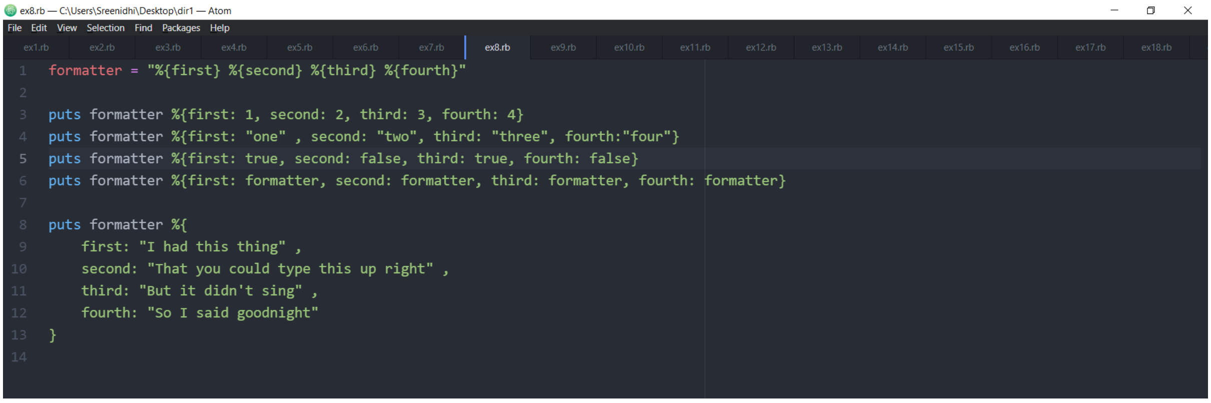

ex7.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb ex8.rb ex9.rb
ex7.rb 1:1
Type here to search

```ruby
puts "Mary had a little lamb."
puts "Its fleece was white as #{'snow'}."
puts "And everywhere that Mary went."
puts "." *10 #what'd you do?

end1 = "C"
end2 = "h"
end3 = "e"
end4 = "e"
end5 = "s"
end6 = "e"
end7 = "B"
end8 = "u"
end9 = "r"
end10 = "g"
end11 = "e"
end12 = "r"

# watch that print vs. puts on this line what's it do?

print end1 + end2 + end3 + end4 + end5 + end6
puts end7 + end8+ end9 + end10 + end11 + end12

# puts includes endline print doesn't

```

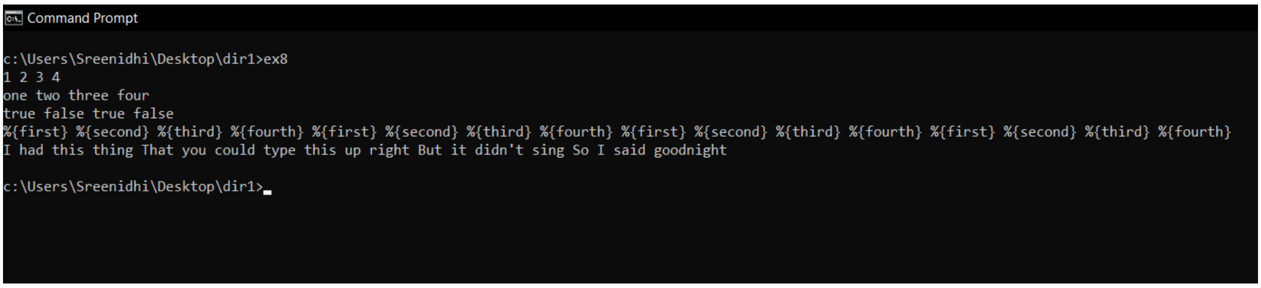


```
c:\Users\Sreenidhi\Desktop\dir1>ex7
Mary had a little lamb.
Its fleece was white as snow.
And everywhere that Mary went.
..........
CheeseBurger

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 8

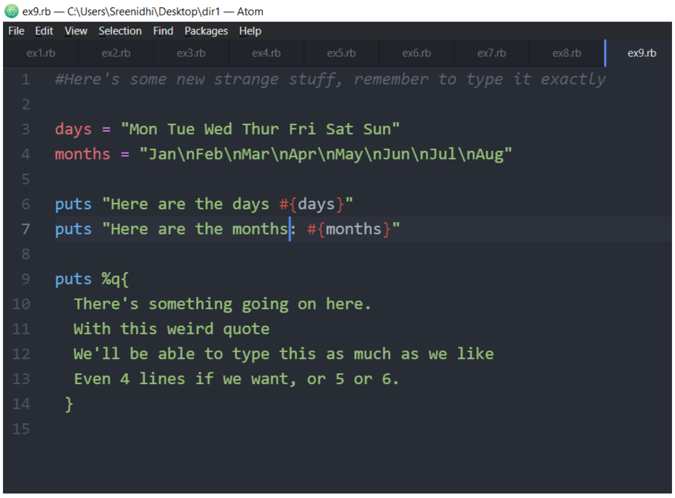

ex8.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb ex8.rb ex9.rb ex10.rb ex11.rb ex12.rb ex13.rb ex14.rb ex15.rb ex16.rb ex17.rb ex18.rb

```ruby
formatter = "%{first} %{second} %{third} %{fourth}"

puts formatter %{first: 1, second: 2, third: 3, fourth: 4}
puts formatter %{first: "one" , second: "two", third: "three", fourth:"four"}
puts formatter %{first: true, second: false, third: true, fourth: false}
puts formatter %{first: formatter, second: formatter, third: formatter, fourth: formatter}

puts formatter %{
    first: "I had this thing" ,
    second: "That you could type this up right" ,
    third: "But it didn't sing" ,
    fourth: "So I said goodnight"
}

```

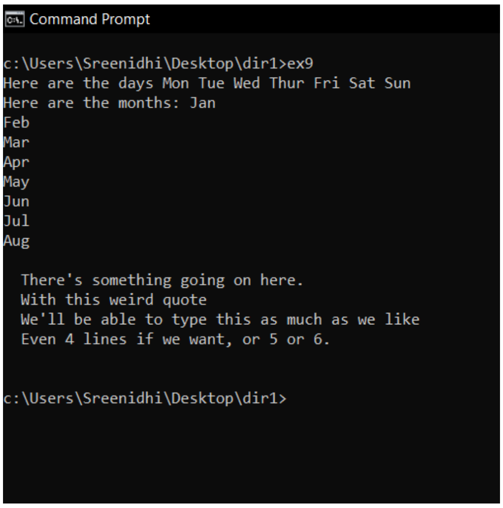


```
c:\Users\Sreenidhi\Desktop\dir1>ex8
1 2 3 4
one two three four
true false true false
%{first} %{second} %{third} %{fourth} %{first} %{second} %{third} %{fourth} %{first} %{second} %{third} %{fourth} %{first} %{second} %{third} %{fourth}
I had this thing That you could type this up right But it didn't sing So I said goodnight

c:\Users\Sreenidhi\Desktop\dir1>_
```

Exercise 9

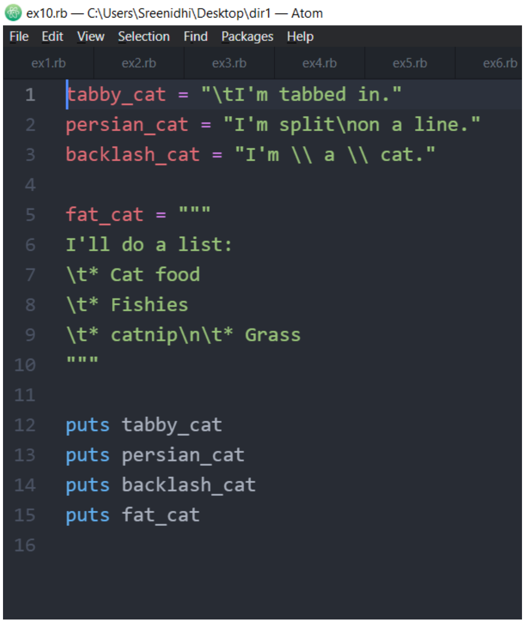

ex9.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb ex8.rb ex9.rb

```ruby
#Here's some new strange stuff, remember to type it exactly

days = "Mon Tue Wed Thur Fri Sat Sun"
months = "Jan\nFeb\nMar\nApr\nMay\nJun\nJul\nAug"

puts "Here are the days #{days}"
puts "Here are the months: #{months}"

puts %q{
  There's something going on here.
  With this weird quote
  We'll be able to type this as much as we like
  Even 4 lines if we want, or 5 or 6.
 }

```

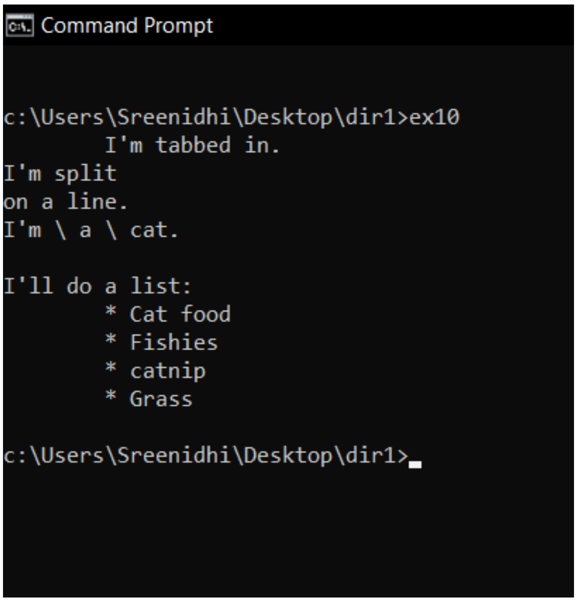


```
c:\Users\Sreenidhi\Desktop\dir1>ex9
Here are the days Mon Tue Wed Thur Fri Sat Sun
Here are the months: Jan
Feb
Mar
Apr
May
Jun
Jul
Aug

  There's something going on here.
  With this weird quote
  We'll be able to type this as much as we like
  Even 4 lines if we want, or 5 or 6.


c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 10

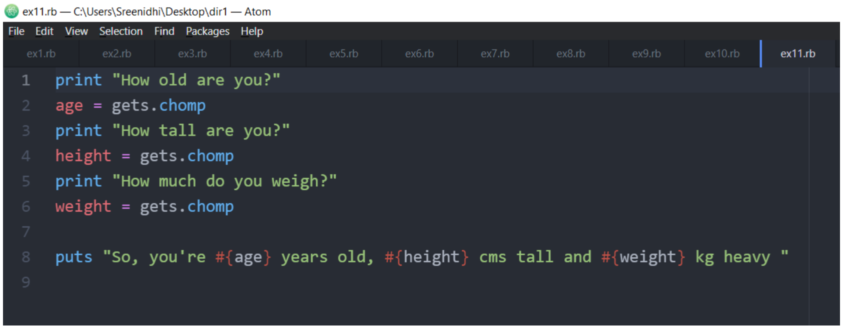

ex10.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb

```ruby
tabby_cat = "\tI'm tabbed in."
persian_cat = "I'm split\non a line."
backlash_cat = "I'm \\ a \\ cat."

fat_cat = """
I'll do a list:
\t* Cat food
\t* Fishies
\t* catnip\n\t* Grass
"""

puts tabby_cat
puts persian_cat
puts backlash_cat
puts fat_cat

```

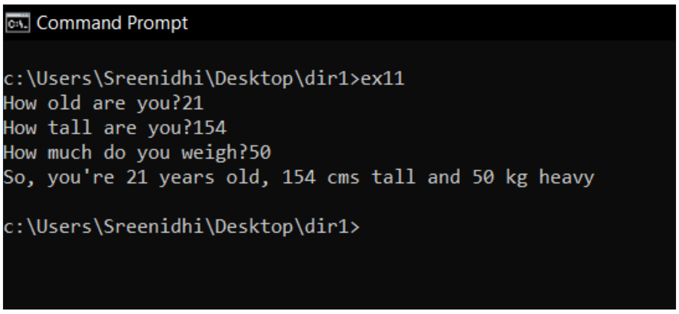


```
c:\Users\Sreenidhi\Desktop\dir1>ex10
        I'm tabbed in.
I'm split
on a line.
I'm \ a \ cat.

I'll do a list:
        * Cat food
        * Fishies
        * catnip
        * Grass

c:\Users\Sreenidhi\Desktop\dir1>_
```

Exercise 11

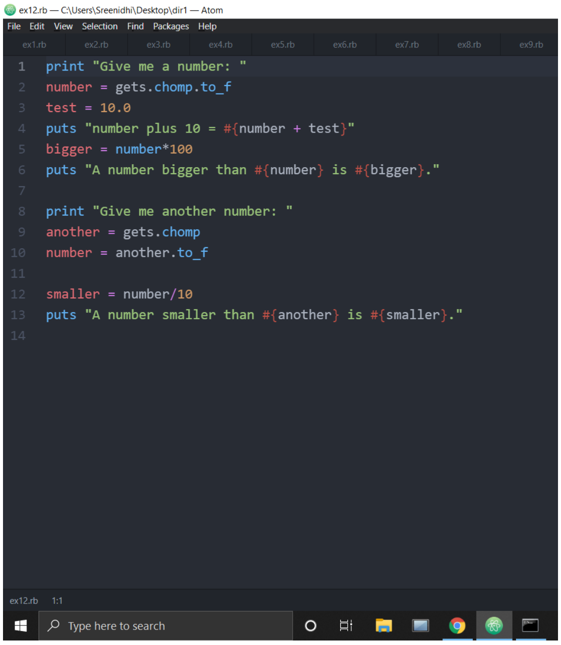

ex11.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb ex8.rb ex9.rb ex10.rb ex11.rb

```ruby
print "How old are you?"
age = gets.chomp
print "How tall are you?"
height = gets.chomp
print "How much do you weigh?"
weight = gets.chomp

puts "So, you're #{age} years old, #{height} cms tall and #{weight} kg heavy "

```

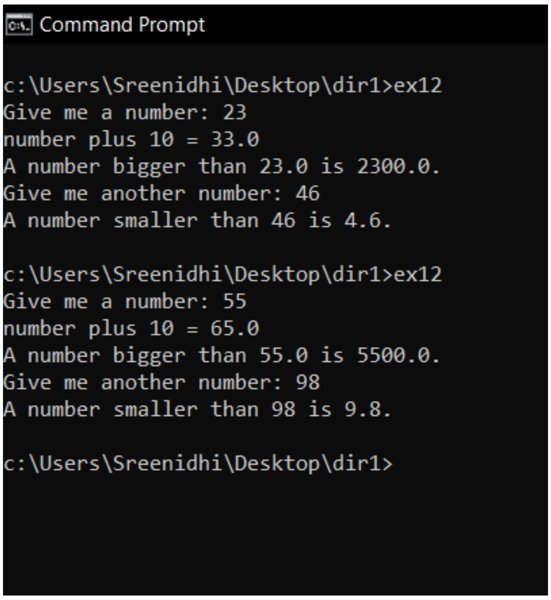


```
c:\Users\Sreenidhi\Desktop\dir1>ex11
How old are you?21
How tall are you?154
How much do you weigh?50
So, you're 21 years old, 154 cms tall and 50 kg heavy

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 12

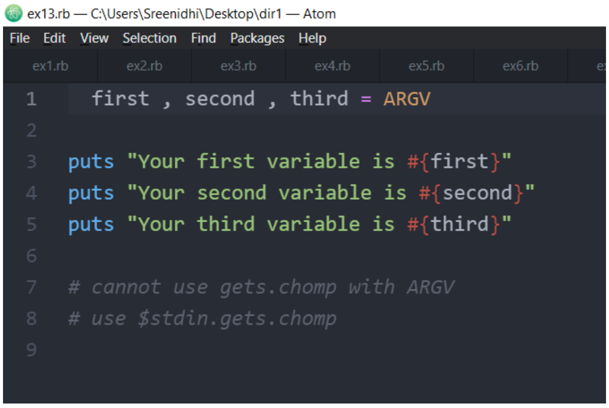

ex12.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb ex8.rb ex9.rb
ex12.rb 1:1
Type here to search

```ruby
print "Give me a number: "
number = gets.chomp.to_f
test = 10.0
puts "number plus 10 = #{number + test}"
bigger = number*100
puts "A number bigger than #{number} is #{bigger}."

print "Give me another number: "
another = gets.chomp
number = another.to_f

smaller = number/10
puts "A number smaller than #{another} is #{smaller}."

```

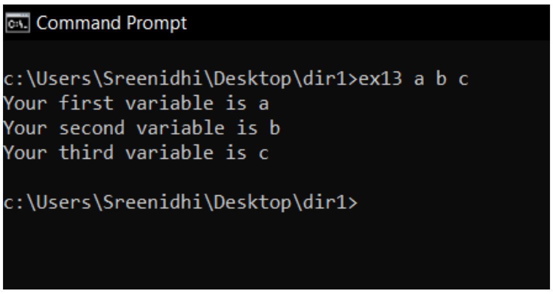


```
c:\Users\Sreenidhi\Desktop\dir1>ex12
Give me a number: 23
number plus 10 = 33.0
A number bigger than 23.0 is 2300.0.
Give me another number: 46
A number smaller than 46 is 4.6.

c:\Users\Sreenidhi\Desktop\dir1>ex12
Give me a number: 55
number plus 10 = 65.0
A number bigger than 55.0 is 5500.0.
Give me another number: 98
A number smaller than 98 is 9.8.

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 13

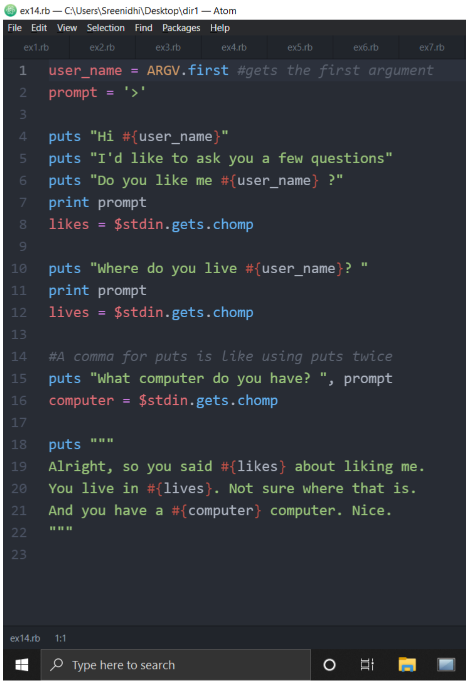

ex13.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb

```ruby
first , second , third = ARGV

puts "Your first variable is #{first}"
puts "Your second variable is #{second}"
puts "Your third variable is #{third}"

# cannot use gets.chomp with ARGV
# use $stdin.gets.chomp
```

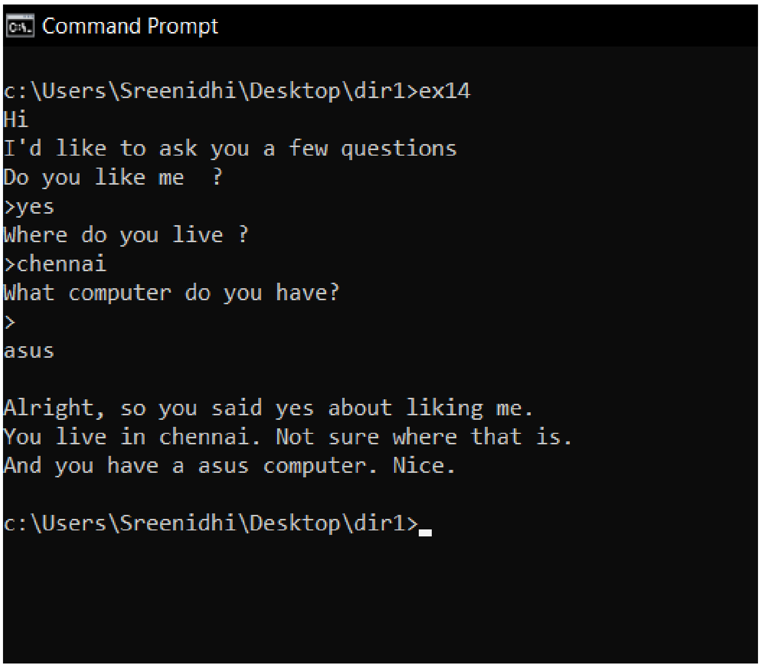


```
c:\Users\Sreenidhi\Desktop\dir1>ex13 a b c
Your first variable is a
Your second variable is b
Your third variable is c

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 14

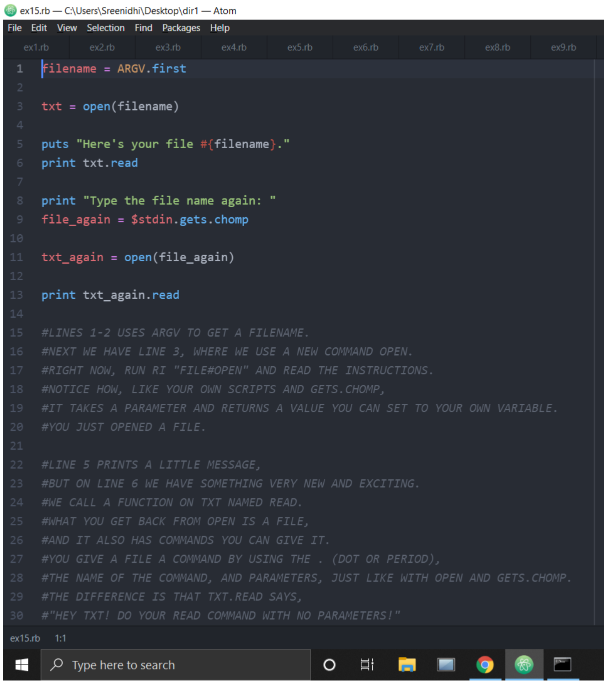

ex14.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb
ex14.rb 1:1
Type here to search

```ruby
user_name = ARGV.first #gets the first argument
prompt = '>'

puts "Hi #{user_name}"
puts "I'd like to ask you a few questions"
puts "Do you like me #{user_name} ?"
print prompt
likes = $stdin.gets.chomp

puts "Where do you live #{user_name}? "
print prompt
lives = $stdin.gets.chomp

#A comma for puts is like using puts twice
puts "What computer do you have? ", prompt
computer = $stdin.gets.chomp

puts """
Alright, so you said #{likes} about liking me.
You live in #{lives}. Not sure where that is.
And you have a #{computer} computer. Nice.
"""
```

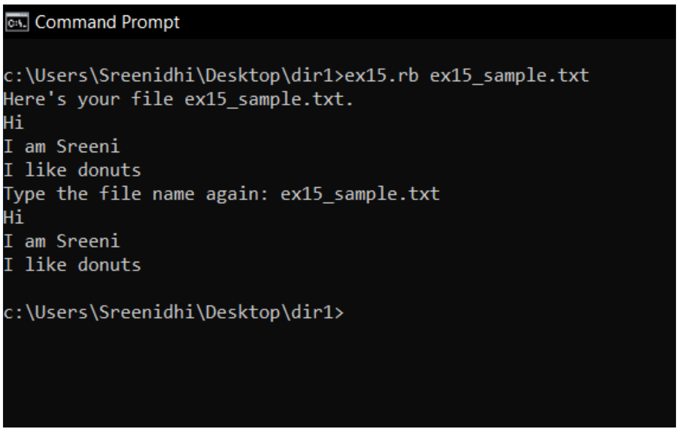


```
c:\Users\Sreenidhi\Desktop\dir1>ex14
Hi
I'd like to ask you a few questions
Do you like me  ?
>yes
Where do you live ?
>chennai
What computer do you have?
>
asus

Alright, so you said yes about liking me.
You live in chennai. Not sure where that is.
And you have a asus computer. Nice.

c:\Users\Sreenidhi\Desktop\dir1>_
```

Exercise 15

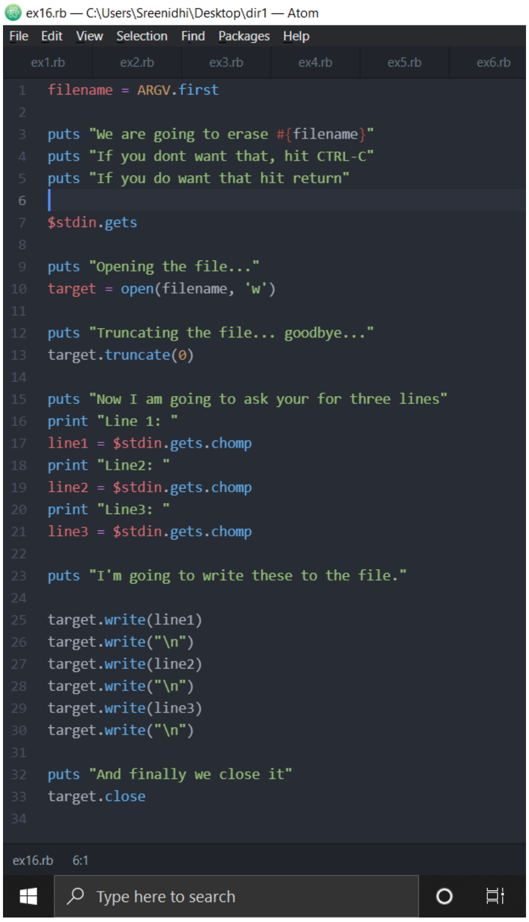

ex15.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb ex8.rb ex9.rb
ex15.rb 1:1
Type here to search

```ruby
1  filename = ARGV.first
2
3  txt = open(filename)
4
5  puts "Here's your file #{filename}."
6  print txt.read
7
8  print "Type the file name again: "
9  file_again = $stdin.gets.chomp
10
11 txt_again = open(file_again)
12
13 print txt_again.read
14
15 #LINES 1-2 USES ARGV TO GET A FILENAME.
16 #NEXT WE HAVE LINE 3, WHERE WE USE A NEW COMMAND OPEN.
17 #RIGHT NOW, RUN RI "FILE#OPEN" AND READ THE INSTRUCTIONS.
18 #NOTICE HOW, LIKE YOUR OWN SCRIPTS AND GETS.CHOMP,
19 #IT TAKES A PARAMETER AND RETURNS A VALUE YOU CAN SET TO YOUR OWN VARIABLE.
20 #YOU JUST OPENED A FILE.
21
22 #LINE 5 PRINTS A LITTLE MESSAGE,
23 #BUT ON LINE 6 WE HAVE SOMETHING VERY NEW AND EXCITING.
24 #WE CALL A FUNCTION ON TXT NAMED READ.
25 #WHAT YOU GET BACK FROM OPEN IS A FILE,
26 #AND IT ALSO HAS COMMANDS YOU CAN GIVE IT.
27 #YOU GIVE A FILE A COMMAND BY USING THE . (DOT OR PERIOD),
28 #THE NAME OF THE COMMAND, AND PARAMETERS, JUST LIKE WITH OPEN AND GETS.CHOMP.
29 #THE DIFFERENCE IS THAT TXT.READ SAYS,
30 #"HEY TXT! DO YOUR READ COMMAND WITH NO PARAMETERS!"
```

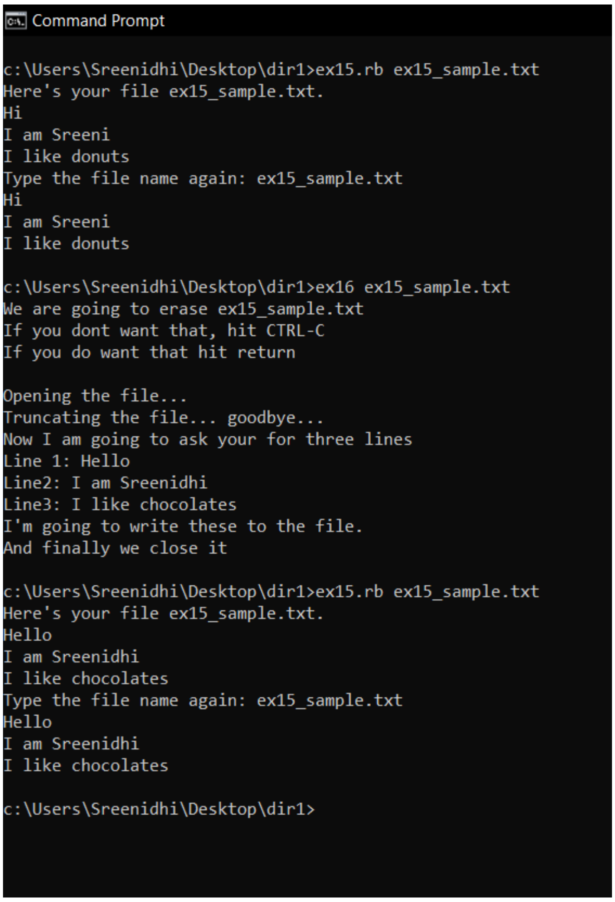


```
c:\Users\Sreenidhi\Desktop\dir1>ex15.rb ex15_sample.txt
Here's your file ex15_sample.txt.
Hi
I am Sreeni
I like donuts
Type the file name again: ex15_sample.txt
Hi
I am Sreeni
I like donuts

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 16

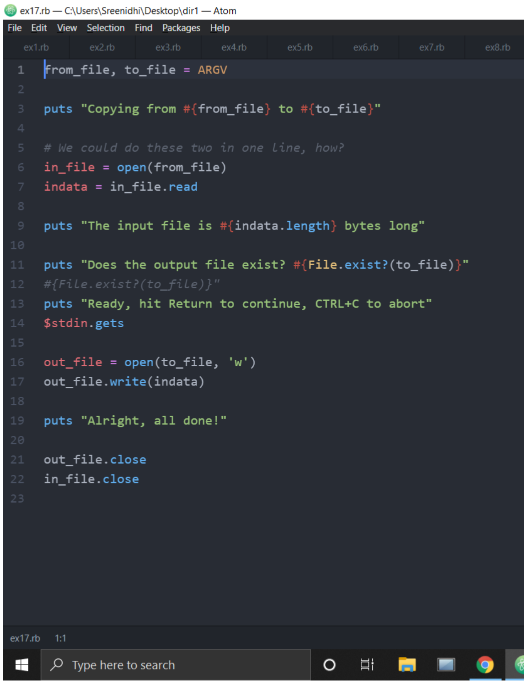

ex16.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb
ex16.rb 6:1
Type here to search

```ruby
filename = ARGV.first

puts "We are going to erase #{filename}"
puts "If you dont want that, hit CTRL-C"
puts "If you do want that hit return"

$stdin.gets

puts "Opening the file..."
target = open(filename, 'w')

puts "Truncating the file... goodbye..."
target.truncate(0)

puts "Now I am going to ask your for three lines"
print "Line 1: "
line1 = $stdin.gets.chomp
print "Line2: "
line2 = $stdin.gets.chomp
print "Line3: "
line3 = $stdin.gets.chomp

puts "I'm going to write these to the file."

target.write(line1)
target.write("\n")
target.write(line2)
target.write("\n")
target.write(line3)
target.write("\n")

puts "And finally we close it"
target.close

```

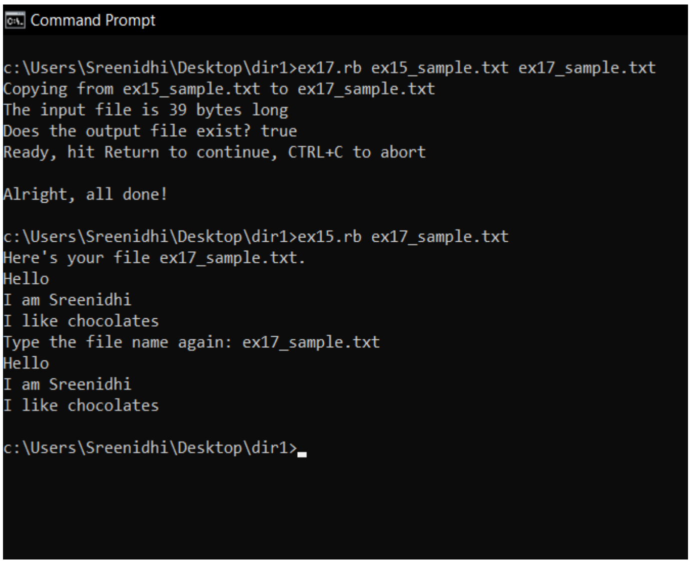


```
c:\Users\Sreenidhi\Desktop\dir1>ex15.rb ex15_sample.txt
Here's your file ex15_sample.txt.
Hi
I am Sreeni
I like donuts
Type the file name again: ex15_sample.txt
Hi
I am Sreeni
I like donuts

c:\Users\Sreenidhi\Desktop\dir1>ex16 ex15_sample.txt
We are going to erase ex15_sample.txt
If you dont want that, hit CTRL-C
If you do want that hit return

Opening the file...
Truncating the file... goodbye...
Now I am going to ask your for three lines
Line 1: Hello
Line2: I am Sreenidhi
Line3: I like chocolates
I'm going to write these to the file.
And finally we close it

c:\Users\Sreenidhi\Desktop\dir1>ex15.rb ex15_sample.txt
Here's your file ex15_sample.txt.
Hello
I am Sreenidhi
I like chocolates
Type the file name again: ex15_sample.txt
Hello
I am Sreenidhi
I like chocolates

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 17

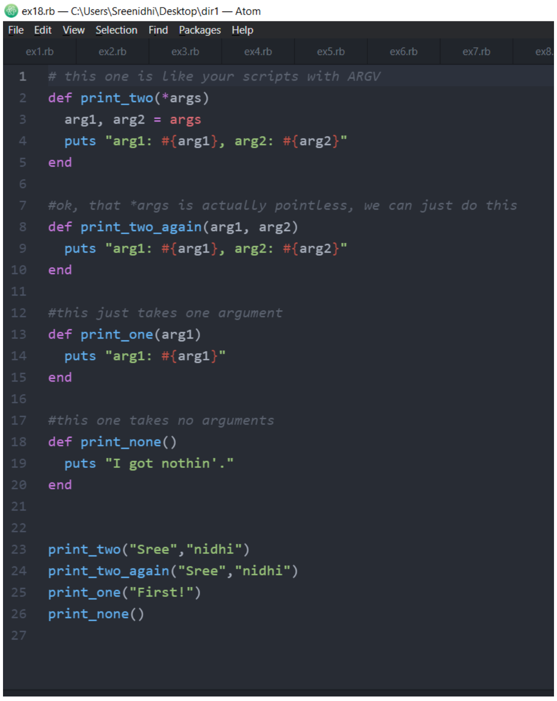

ex17.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb ex2.rb ex3.rb ex4.rb ex5.rb ex6.rb ex7.rb ex8.rb
ex17.rb 1:1
Type here to search

```ruby
from_file, to_file = ARGV

puts "Copying from #{from_file} to #{to_file}"

# We could do these two in one line, how?
in_file = open(from_file)
indata = in_file.read

puts "The input file is #{indata.length} bytes long"

puts "Does the output file exist? #{File.exist?(to_file)}"
#{File.exist?(to_file)}"
puts "Ready, hit Return to continue, CTRL+C to abort"
$stdin.gets

out_file = open(to_file, 'w')
out_file.write(indata)

puts "Alright, all done!"

out_file.close
in_file.close

```

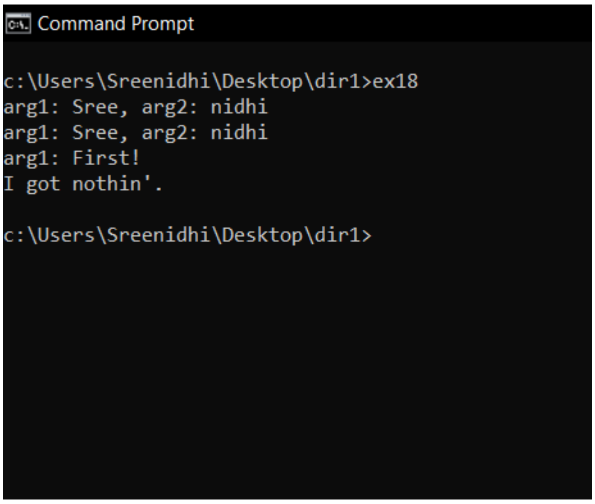


```
c:\Users\Sreenidhi\Desktop\dir1>ex17.rb ex15_sample.txt ex17_sample.txt
Copying from ex15_sample.txt to ex17_sample.txt
The input file is 39 bytes long
Does the output file exist? true
Ready, hit Return to continue, CTRL+C to abort

Alright, all done!

c:\Users\Sreenidhi\Desktop\dir1>ex15.rb ex17_sample.txt
Here's your file ex17_sample.txt.
Hello
I am Sreenidhi
I like chocolates
Type the file name again: ex17_sample.txt
Hello
I am Sreenidhi
I like chocolates

c:\Users\Sreenidhi\Desktop\dir1>_
```

Exercise 18

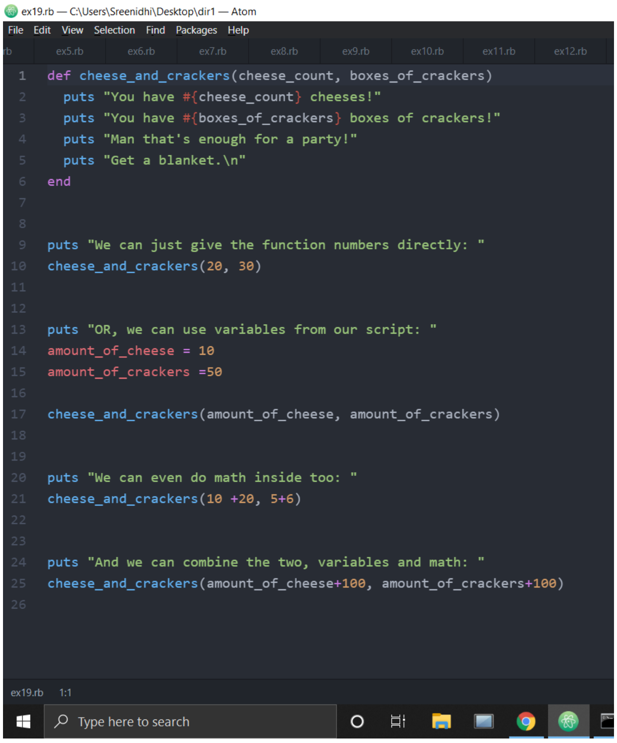

ex18.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
ex1.rb
ex2.rb
ex3.rb
ex4.rb
ex5.rb
ex6.rb
ex7.rb
ex8

```ruby
# this one is like your scripts with ARGV
def print_two(*args)
  arg1, arg2 = args
  puts "arg1: #{arg1}, arg2: #{arg2}"
end

#ok, that *args is actually pointless, we can just do this
def print_two_again(arg1, arg2)
  puts "arg1: #{arg1}, arg2: #{arg2}"
end

#this just takes one argument
def print_one(arg1)
  puts "arg1: #{arg1}"
end

#this one takes no arguments
def print_none()
  puts "I got nothin'."
end


print_two("Sree","nidhi")
print_two_again("Sree","nidhi")
print_one("First!")
print_none()
```

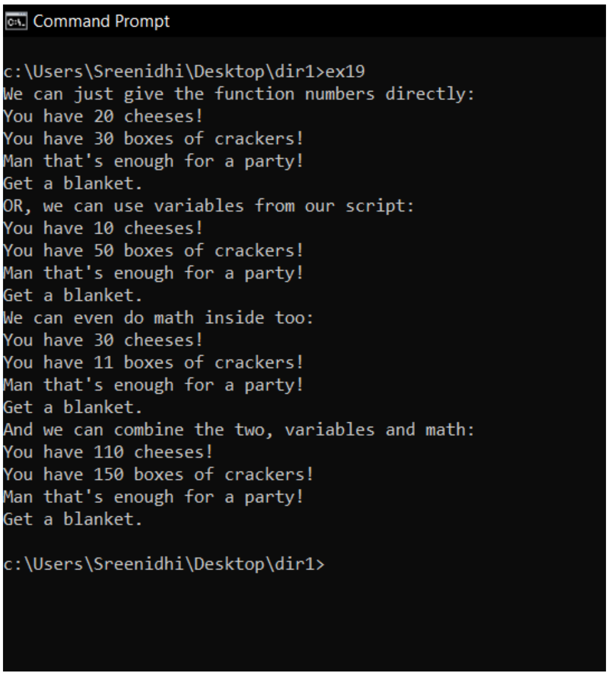


```
c:\Users\Sreenidhi\Desktop\dir1>ex18
arg1: Sree, arg2: nidhi
arg1: Sree, arg2: nidhi
arg1: First!
I got nothin'.

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 19

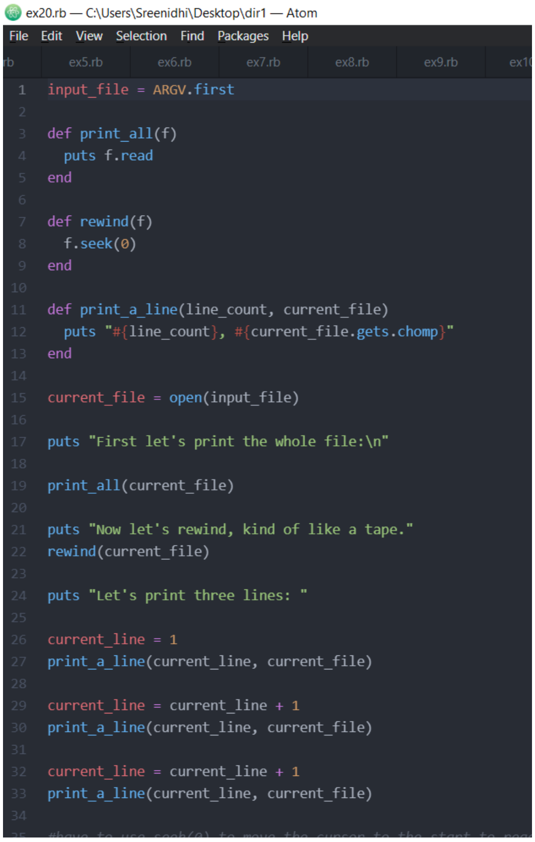

ex19.rb — C:\Users\Sreenidhi\Desktop\dir1 — Atom
File Edit View Selection Find Packages Help
rb
ex5.rb
ex6.rb
ex7.rb
ex8.rb
ex9.rb
ex10.rb
ex11.rb
ex12.rb
ex19.rb 1:1
Type here to search

```ruby
def cheese_and_crackers(cheese_count, boxes_of_crackers)
  puts "You have #{cheese_count} cheeses!"
  puts "You have #{boxes_of_crackers} boxes of crackers!"
  puts "Man that's enough for a party!"
  puts "Get a blanket.\n"
end


puts "We can just give the function numbers directly: "
cheese_and_crackers(20, 30)


puts "OR, we can use variables from our script: "
amount_of_cheese = 10
amount_of_crackers =50

cheese_and_crackers(amount_of_cheese, amount_of_crackers)


puts "We can even do math inside too: "
cheese_and_crackers(10 +20, 5+6)


puts "And we can combine the two, variables and math: "
cheese_and_crackers(amount_of_cheese+100, amount_of_crackers+100)
```

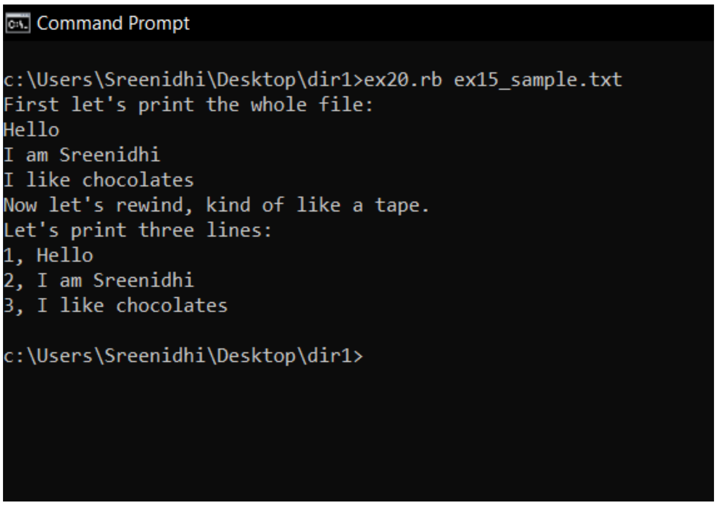


```
c:\Users\Sreenidhi\Desktop\dir1>ex19
We can just give the function numbers directly:
You have 20 cheeses!
You have 30 boxes of crackers!
Man that's enough for a party!
Get a blanket.
OR, we can use variables from our script:
You have 10 cheeses!
You have 50 boxes of crackers!
Man that's enough for a party!
Get a blanket.
We can even do math inside too:
You have 30 cheeses!
You have 11 boxes of crackers!
Man that's enough for a party!
Get a blanket.
And we can combine the two, variables and math:
You have 110 cheeses!
You have 150 boxes of crackers!
Man that's enough for a party!
Get a blanket.

c:\Users\Sreenidhi\Desktop\dir1>
```

Exercise 20



```ruby
input_file = ARGV.first

def print_all(f)
  puts f.read
end

def rewind(f)
  f.seek(0)
end

def print_a_line(line_count, current_file)
  puts "#{line_count}, #{current_file.gets.chomp}"
end

current_file = open(input_file)

puts "First let's print the whole file:\n"

print_all(current_file)

puts "Now let's rewind, kind of like a tape."
rewind(current_file)

puts "Let's print three lines: "

current_line = 1
print_a_line(current_line, current_file)

current_line = current_line + 1
print_a_line(current_line, current_file)

current_line = current_line + 1
print_a_line(current_line, current_file)

#have to use seek(0) to move the cursor to the start to rea
```



```
c:\Users\Sreenidhi\Desktop\dir1>ex20.rb ex15_sample.txt
First let's print the whole file:
Hello
I am Sreenidhi
I like chocolates
Now let's rewind, kind of like a tape.
Let's print three lines:
1, Hello
2, I am Sreenidhi
3, I like chocolates

c:\Users\Sreenidhi\Desktop\dir1>
```