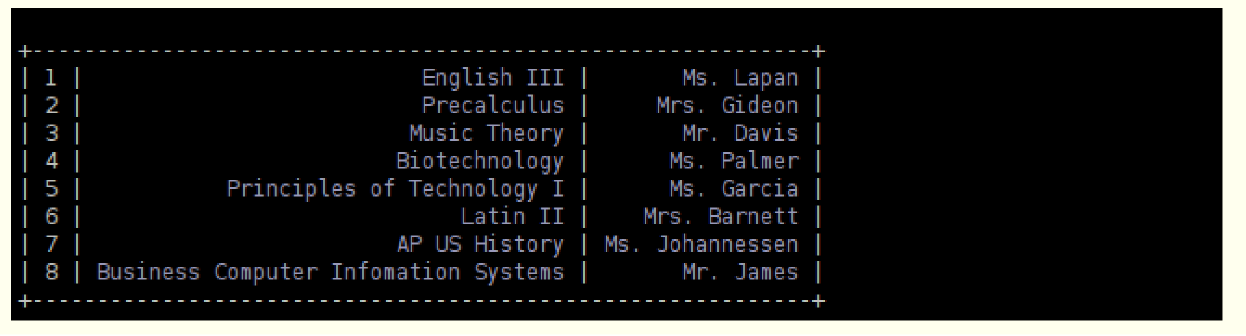
P1. Your Schedule

Use several variables to store the names of your classes and their teachers. Then, display a nice little table displaying your schedule. Just FYI, my column of courses has a width of 26 characters, and the teacher column has a width of 15. The first and last lines are a plus sign, fifty dashes (a.k.a. minus signs) and another plus sign.

Your table doesn't need to look exactly like this, or even line up. I used a total of sixteen variables (course1, course2, ... course8, teacher1, teacher2, etc.). You should do the same.

Example



```
+------------------------------------------------------------+
| 1 |                          English III |       Ms. Lapan |
| 2 |                          Precalculus |     Mrs. Gideon |
| 3 |                         Music Theory |       Mr. Davis |
| 4 |                        Biotechnology |      Ms. Palmer |
| 5 |           Principles of Technology I |      Ms. Garcia |
| 6 |                             Latin II |    Mrs. Barnett |
| 7 |                        AP US History | Ms. Johannessen |
| 8 | Business Computer Infomation Systems |       Mr. James |
+------------------------------------------------------------+
```

P2. Asking Questions

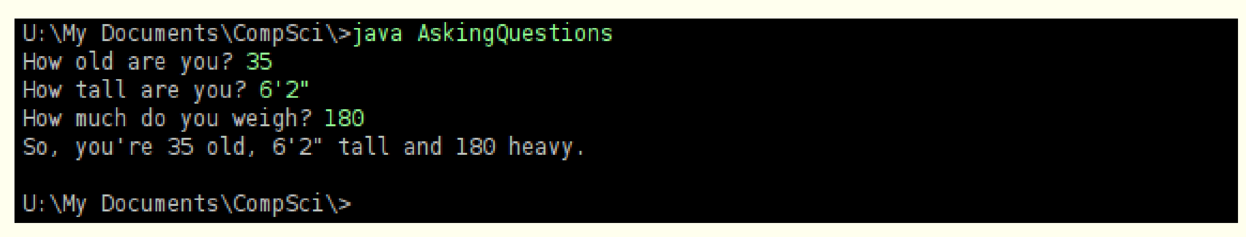
It's now time to pick up the pace a bit. I've got you doing a lot of printing so that you get used to typing simple things, but those simple things are fairly boring. What we want to do now is get you getting data into your programs. This though is a little tricky so we have to have you learn to do two things that may not make sense right away, but if you stick with it they should click suddenly a few exercises from now.

Most of what software does is the following:

1. Take some kind of input from a person.
2. Change it.
3. Print out something to show how it changed.

So far you've only been printing, but you haven't been able to get any input from a person, or change it. You may not even know what "input" means, so rather than talk about it, let's have you do some and see if you get it. Next exercise we'll do more to explain it.

Example

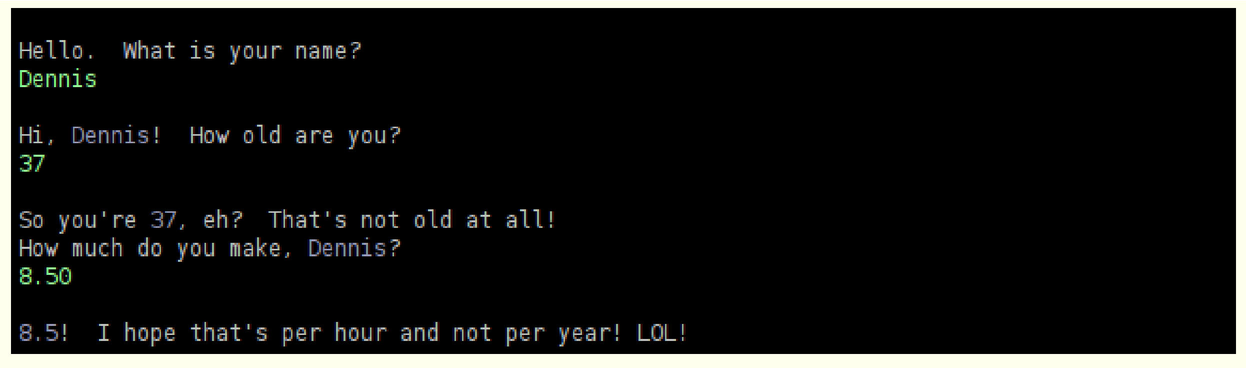


```
U:\My Documents\CompSci\>java AskingQuestions
How old are you? 35
How tall are you? 6'2"
How much do you weigh? 180
So, you're 35 old, 6'2" tall and 180 heavy.

U:\My Documents\CompSci\>
```

P3. Name, Age, and Salary

Ask the user for their name. Then display their name to prove that you can recall it. Ask them for their age. Then display that. Finally, ask them for how much they make and display that. You should use the most appropriate data type for each variable.

Example



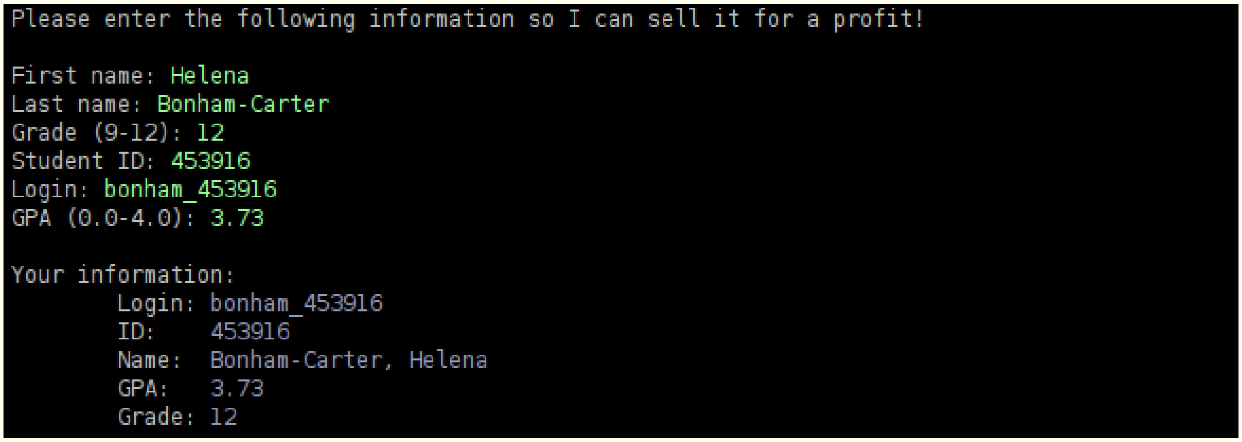
```

Hello.  What is your name?
Dennis

Hi, Dennis!  How old are you?
37

So you're 37, eh?  That's not old at all!
How much do you make, Dennis?
8.50

8.5!  I hope that's per hour and not per year! LOL!
```

P4. More User Input of Data

Ask the user for several pieces of information, and display them on the screen afterward as a summary.

- first name
- last name
- grade (classification)
- student id number
- login name
- GPA (0.0 to 4.0)

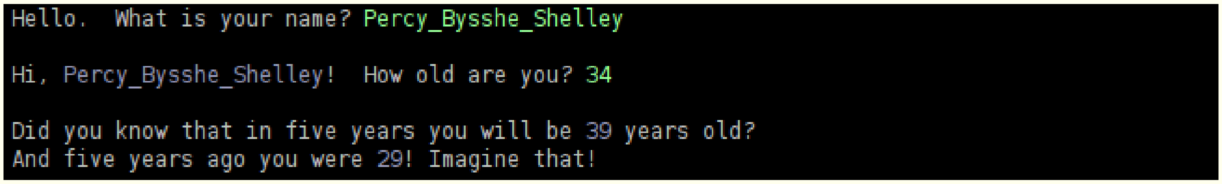
You must use the most appropriate type for each variable and not just Strings for everything. Use user validations wherever appropriate.



```
Please enter the following information so I can sell it for a profit!

First name: Helena
Last name: Bonham-Carter
Grade (9-12): 12
Student ID: 453916
Login: bonham_453916
GPA (0.0-4.0): 3.73

Your information:
        Login: bonham_453916
        ID:    453916
        Name:  Bonham-Carter, Helena
        GPA:   3.73
        Grade: 12
```

P5. Age in Five Years

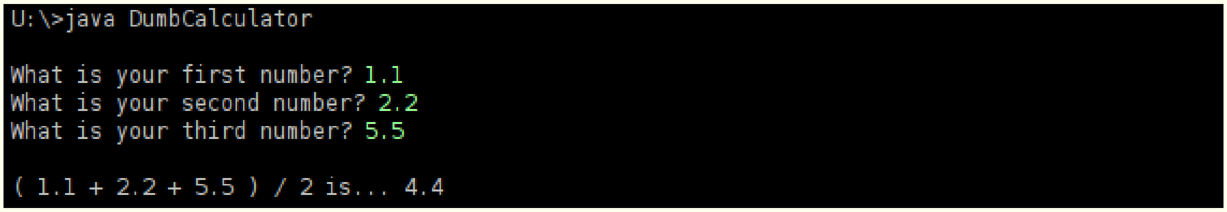
Ask the user for their name. Then display their name to prove that you can recall it. Ask them for their age. Then display what their age would be five years from now. Then display what their age would be five years ago.



```
Hello.  What is your name? Percy_Bysshe_Shelley

Hi, Percy_Bysshe_Shelley!  How old are you? 34

Did you know that in five years you will be 39 years old?
And five years ago you were 29! Imagine that!
```

P6. A Dumb Calculator

Make a simple numeric calculator. It should prompt the user for three numbers. Then add the numbers together and divide by 2. Display the result. Your program must support numbers with decimals and not just integers.

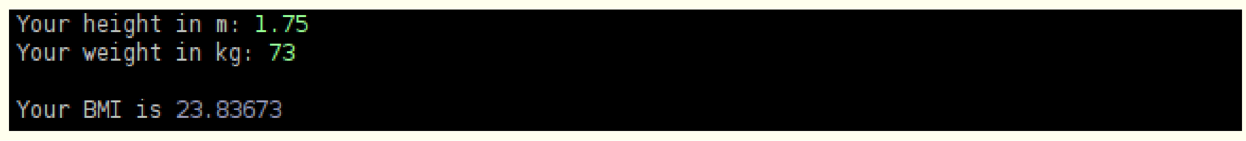


```
U:\>java DumbCalculator

What is your first number? 1.1
What is your second number? 2.2
What is your third number? 5.5

( 1.1 + 2.2 + 5.5 ) / 2 is... 4.4
```

P7. BMI Calculator

The body mass index (BMI) is commonly used by health and nutrition professionals to estimate human body fat in populations.

It is computed by taking the individual's weight (mass) in kilograms and dividing it by the square of their height in meters.



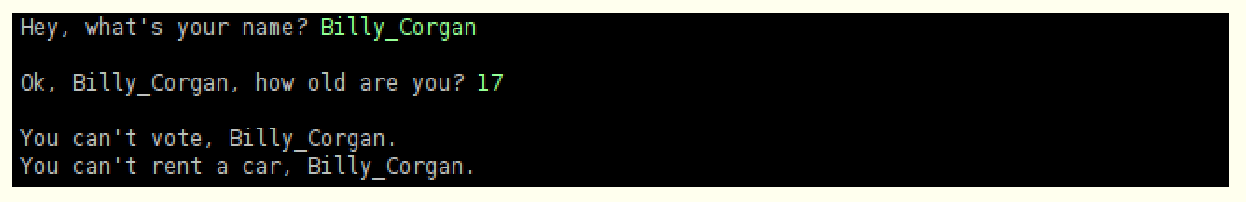
```
Your height in m: 1.75
Your weight in kg: 73

Your BMI is 23.83673
```

P8. How Old Are You?

Make a program which displays a different message depending on the age given. Here are the possible responses:

- age is less than 16, say "You can't drive."
- age is less than 18, say "You can't vote."
- age is less than 25, say "You can't rent a car."
- age is 25 or over, say "You can do anything that's legal."

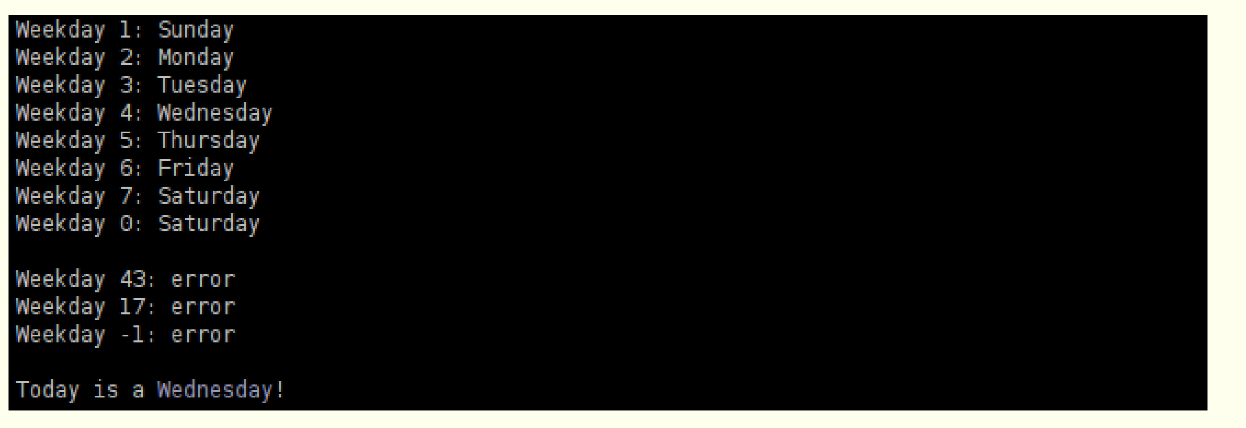
Here's a sample run. Notice that a person who is under 16 will display *three* messages, one for being under 16, one for also being under 18, and one for also being under 25



```
Hey, what's your name? Billy_Corgan

Ok, Billy_Corgan, how old are you? 17

You can't vote, Billy_Corgan.
You can't rent a car, Billy_Corgan.
```

P9. Weekday Name

I have provided a function that is supposed to return the name of a day of the week given the day number.



```
Weekday 1: Sunday
Weekday 2: Monday
Weekday 3: Tuesday
Weekday 4: Wednesday
Weekday 5: Thursday
Weekday 6: Friday
Weekday 7: Saturday
Weekday 0: Saturday

Weekday 43: error
Weekday 17: error
Weekday -1: error

Today is a Wednesday!
```

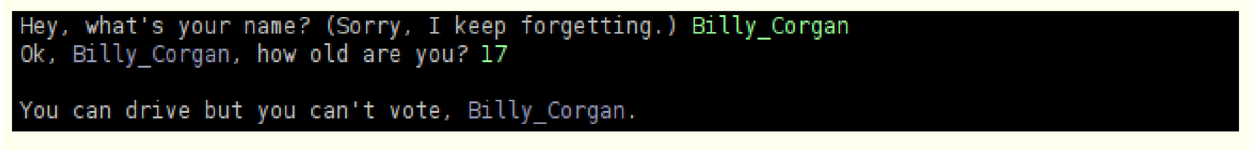
P10. How Old Are You, Specifically?

Using if statements, else if, and else statements, make a program which displays a different message depending on the age given.

age	**message**

less than 16	"You can't drive."
16 to 17	"You can drive but not vote."
18 to 24	"You can vote but not rent a car."
25 or older	"You can do pretty much anything."

Note that unlike the Hoe Old Are You assignment, this program must only display *exactly one* message for a given age and not multiple messages.



```
Hey, what's your name? (Sorry, I keep forgetting.) Billy_Corgan
Ok, Billy_Corgan, how old are you? 17

You can drive but you can't vote, Billy_Corgan.
```

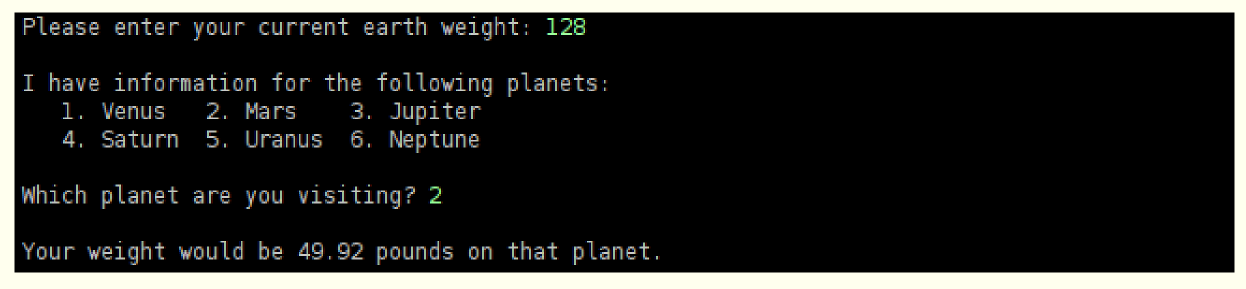
P11. Space Boxing

Julio Cesar Chavez Mark VII is an interplanetary space boxer, who currently holds the championship belts for various weight categories on many different planets within our solar system. However, it is often difficult for him to recall what his "target weight" needs to be on earth in order to make the weight class on other planets.   Write a program to help him keep track of this.

It should ask him what his earth weight is, and to enter a number for the planet he wants to fight on. It should then compute his weight on the destination planet based on the table below:

#	**Planet**	**Relative gravity**
1	Venus	0.78
2	Mars	0.39
3	Jupiter	2.65
4	Saturn	1.17
5	Uranus	1.05
6	Neptune	1.23

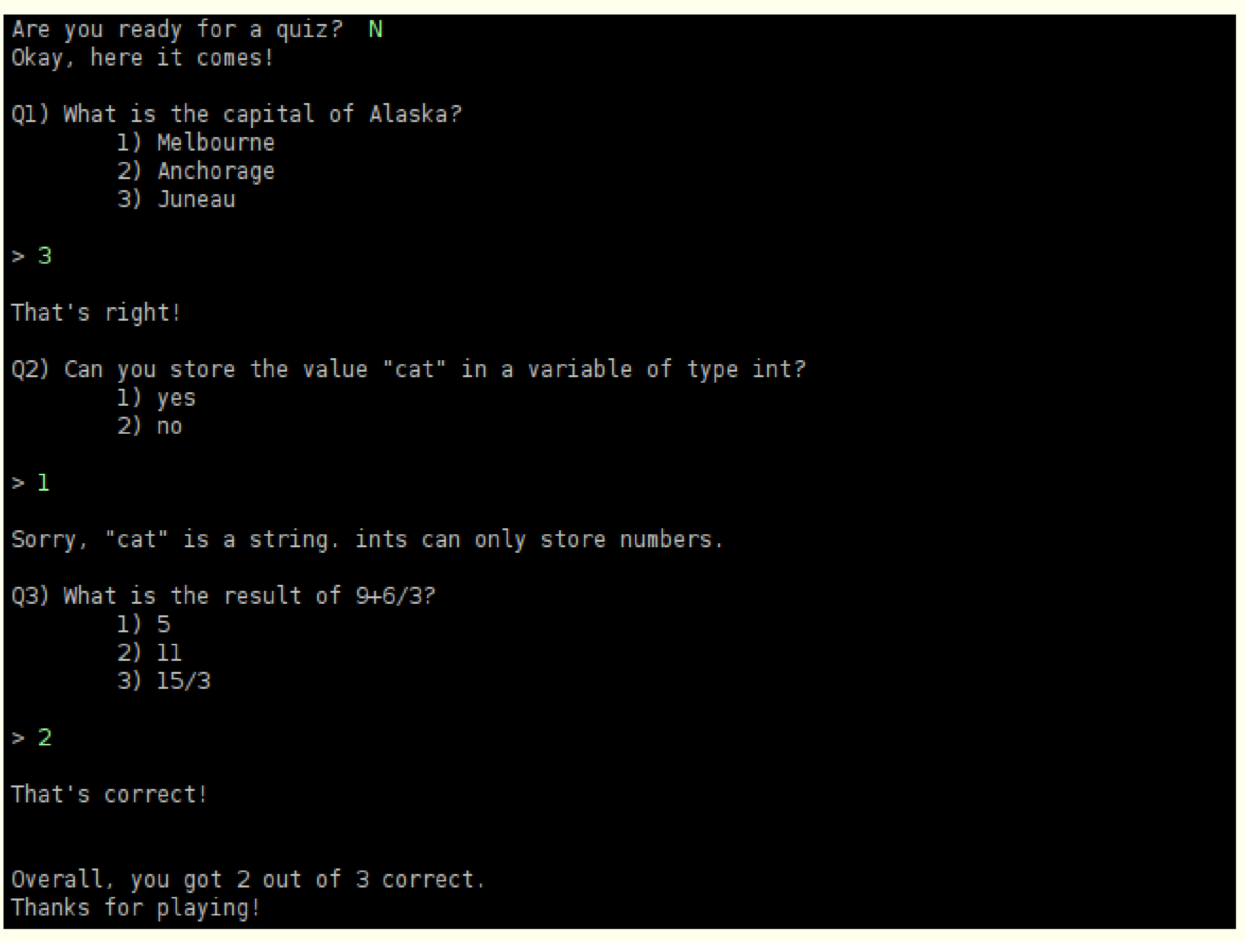
So, for example, if Julio weighs 128 lbs. on earth, then he would weigh just under 50 lbs. on Mars, since Mars' gravity is 0.39 times earth's gravity. (128 * 0.39 is 49.92)



```
Please enter your current earth weight: 128

I have information for the following planets:
   1. Venus   2. Mars    3. Jupiter
   4. Saturn  5. Uranus  6. Neptune

Which planet are you visiting? 2

Your weight would be 49.92 pounds on that planet.
```

P12. A Little Quiz

Write an interactive quiz. It should ask the user three multiple-choice or true/false questions about something. It must keep track of how many they get wrong, and print out a "score" at the end.



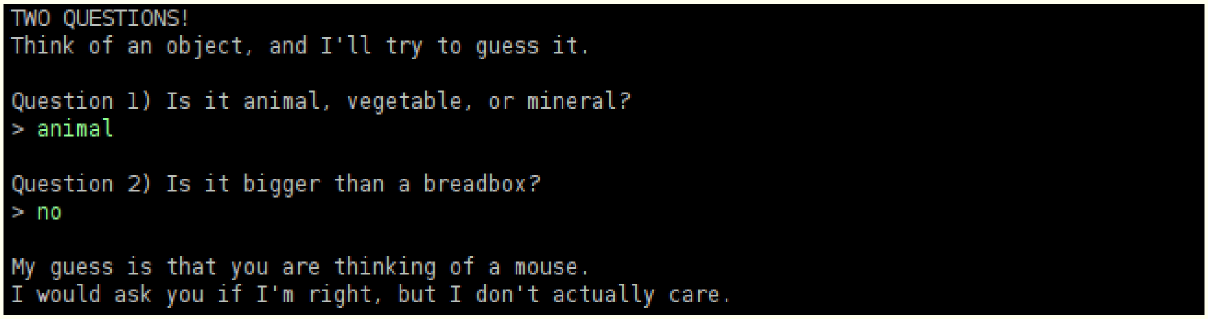
```
Are you ready for a quiz?  N
Okay, here it comes!

Q1) What is the capital of Alaska?
        1) Melbourne
        2) Anchorage
        3) Juneau

> 3

That's right!

Q2) Can you store the value "cat" in a variable of type int?
        1) yes
        2) no

> 1

Sorry, "cat" is a string. ints can only store numbers.

Q3) What is the result of 9+6/3?
        1) 5
        2) 11
        3) 15/3

> 2

That's correct!


Overall, you got 2 out of 3 correct.
Thanks for playing!
```

P13. Make a program which plays a simple game of 20 2 Questions. The first question should be "animal, vegetable, or mineral?" Then, the second question should be "is it bigger than a breadbox?" Then, display one of six possible responses, depending on their answers. You can choose what answers to give for each of the six possibilities.

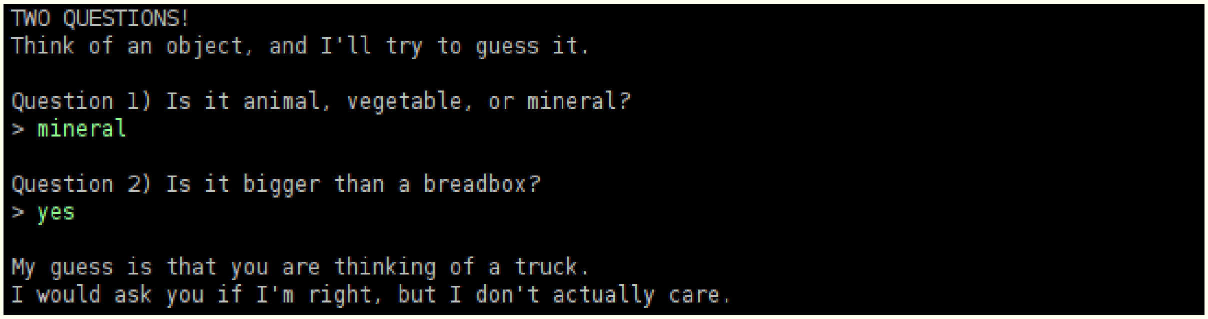
Here's a suggestion:

size \ type	**animal**	**vegetable**	**mineral**
smaller than a breadbox	squirrel	carrot	paper clip
bigger than a breadbox	moose	watermelon	Camaro

You will use *nested* **if** statements to do this.



```
TWO QUESTIONS!
Think of an object, and I'll try to guess it.

Question 1) Is it animal, vegetable, or mineral?
> animal

Question 2) Is it bigger than a breadbox?
> no

My guess is that you are thinking of a mouse.
I would ask you if I'm right, but I don't actually care.
```

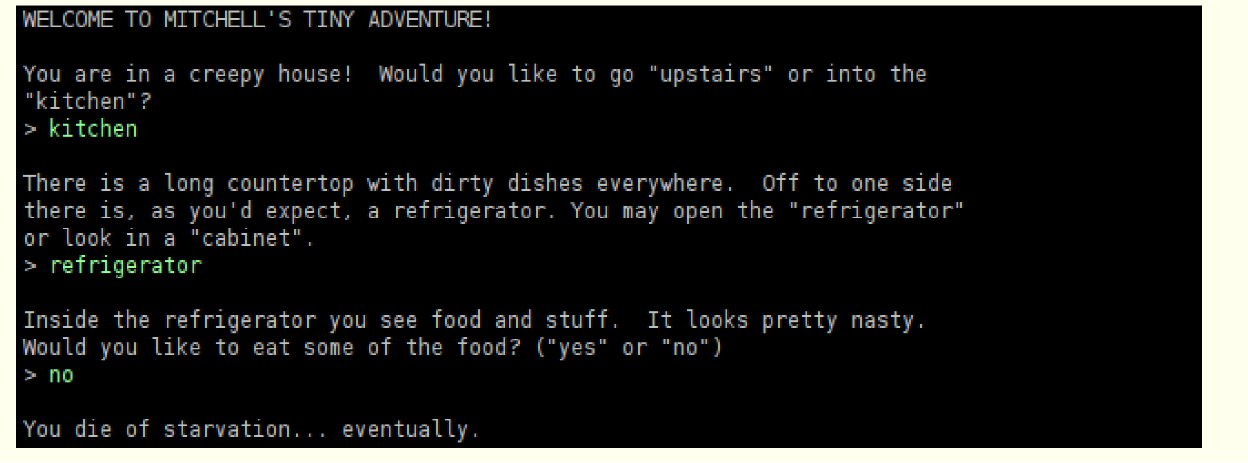


```
TWO QUESTIONS!
Think of an object, and I'll try to guess it.

Question 1) Is it animal, vegetable, or mineral?
> mineral

Question 2) Is it bigger than a breadbox?
> yes

My guess is that you are thinking of a truck.
I would ask you if I'm right, but I don't actually care.
```

P14. **Choose Your Own Adventure!**

```
      _   R1    _
      /            \
   R2             R3
  /    \             /  \
R4   R5       R6    R7
/\       /   \      / \      / \
E1 E2 E3 E4  E5 E6 E7 E8
```

You must use *nested* if statements to do this.



```
WELCOME TO MITCHELL'S TINY ADVENTURE!

You are in a creepy house!  Would you like to go "upstairs" or into the
"kitchen"?
> kitchen

There is a long countertop with dirty dishes everywhere.  Off to one side
there is, as you'd expect, a refrigerator. You may open the "refrigerator"
or look in a "cabinet".
> refrigerator

Inside the refrigerator you see food and stuff.  It looks pretty nasty.
Would you like to eat some of the food? ("yes" or "no")
> no

You die of starvation... eventually.
```

\

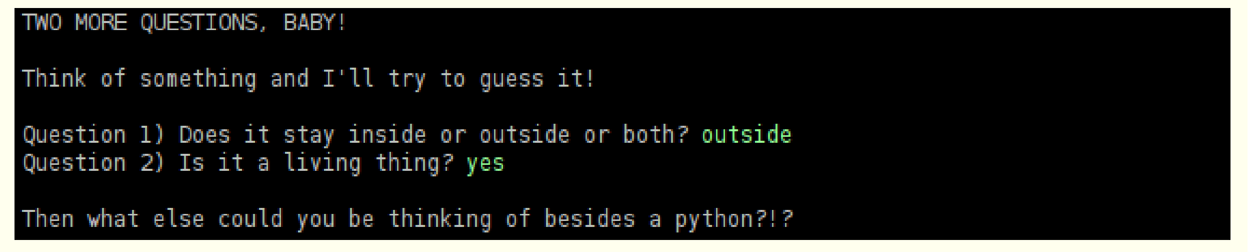
P.15 Two More Questions

Using if statements with compound conditions (like &&), make a guessing game of two questions similar to the Twenty Questions assignment.
However, this time you *must* accomplish it using if statements with compound conditions and you *must not* use else if or else or nested ifs.

- Question 1: Does it belong inside or outside or both?
- Question 2: Is it alive?

Again, here are some sample responses, for the non-creative among you.

	inside	**outside**	**both**
alive	houseplant	bison	dog
not alive	shower curtain	billboard	cell phone

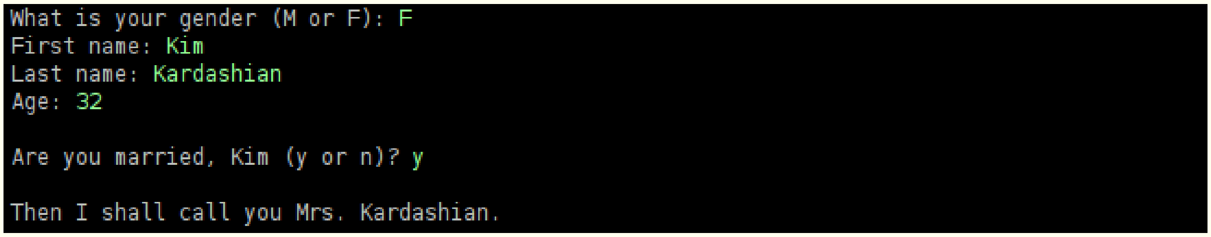


```
TWO MORE QUESTIONS, BABY!

Think of something and I'll try to guess it!

Question 1) Does it stay inside or outside or both? outside
Question 2) Is it a living thing? yes

Then what else could you be thinking of besides a python?!?
```

P16. **Gender Game**

Make a program which displays an appropriate name for a person, using a combination of nested ifs and compound conditions. Ask the user for a gender, first name, last name and age.

If the person is female and 20 or over, ask if she is married. If so, display "Mrs." in front of her name. If not, display "Ms." in front of her name. If the female is under 20, display her first and last name.
If the person is male and 20 or over, display "Mr." in front of his name. Otherwise, display his first and last name.
Note that asking a person if they are married should *only* be done if they are female and 20 or older, which means you will have a single if and else nested inside one of your if statements. Also, did you know that with an if statement (or else), the curly braces are **optional** when there is only one statement inside?

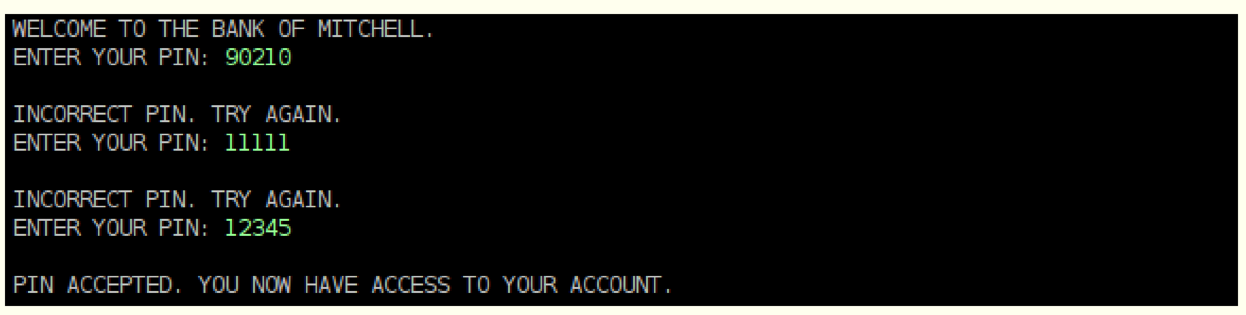


```
What is your gender (M or F): F
First name: Kim
Last name: Kardashian
Age: 32

Are you married, Kim (y or n)? y

Then I shall call you Mrs. Kardashian.
```

P17. **Enter Your PIN**

Understand the below Java program. Write an equivalent code in Ruby for the below Java Program

```
import java.util.Scanner;

public class EnterPIN
{
    public static void main( String[] args )
    {
        Scanner keyboard = new Scanner(System.in);
        int pin = 12345;

        System.out.println("WELCOME TO THE BANK OF MITCHELL.");
        System.out.print("ENTER YOUR PIN: ");
        int entry = keyboard.nextInt();

        while ( entry != pin )
        {
            System.out.println("\nINCORRECT PIN. TRY AGAIN.");
            System.out.print("ENTER YOUR PIN: ");
            entry = keyboard.nextInt();
        }

        System.out.println("\nPIN ACCEPTED. YOU NOW HAVE ACCESS TO YOUR ACCOUNT.");
        }
}
```



```
WELCOME TO THE BANK OF MITCHELL.
ENTER YOUR PIN: 90210

INCORRECT PIN. TRY AGAIN.
ENTER YOUR PIN: 11111

INCORRECT PIN. TRY AGAIN.
ENTER YOUR PIN: 12345

PIN ACCEPTED. YOU NOW HAVE ACCESS TO YOUR ACCOUNT.
```

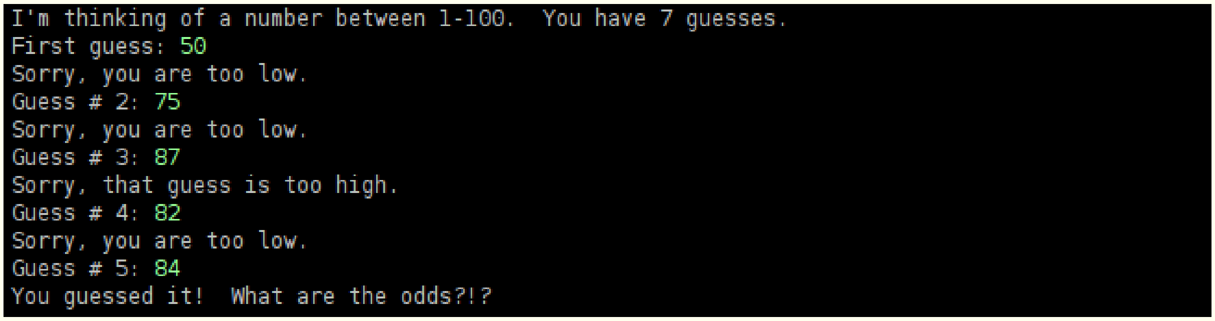
P18. **PIN Lockout**

Understand the below Java program. Write an equivalent code in Ruby for the below Java Program

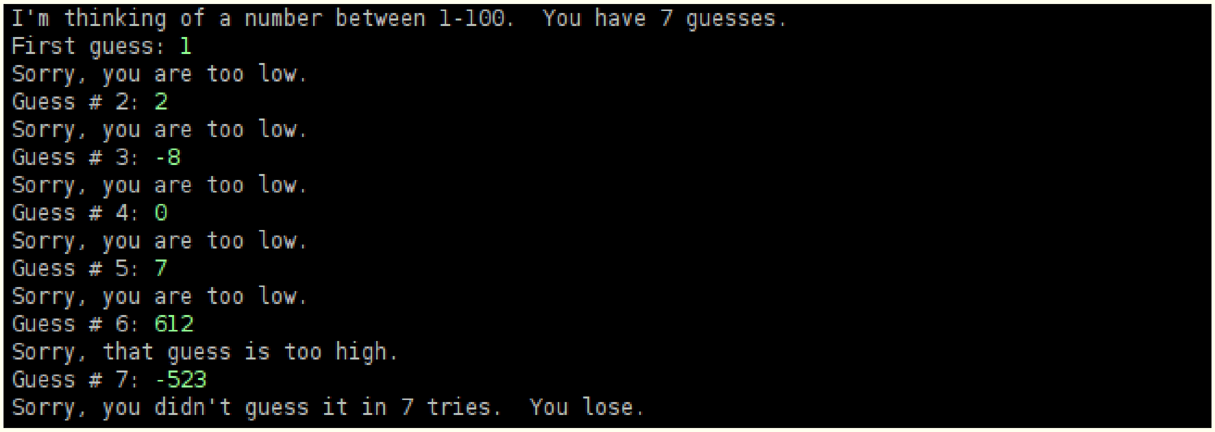
```
import java.util.Scanner;

public class PinLockout
{
        public static void main( String[] args )
        {
                Scanner keyboard = new Scanner(System.in);
                int pin = 12345;
                int tries = 0;

                System.out.println("WELCOME TO THE BANK OF MITCHELL.");
                System.out.print("ENTER YOUR PIN: ");
                int entry = keyboard.nextInt();
                tries++;

                while ( entry != pin && tries < 3 )
                {
                        System.out.println("\nINCORRECT PIN. TRY AGAIN.");
                        System.out.print("ENTER YOUR PIN: ");
                        entry = keyboard.nextInt();
                        tries++;
                }

                if ( entry == pin )
                        System.out.println("\nPIN ACCEPTED. YOU NOW HAVE ACCESS TO
YOUR ACCOUNT.");
                    else if ( tries >= 3 )
                              System.out.println("\nYOU HAVE RUN OUT OF TRIES. ACCOUNT
LOCKED.");
          }
}
```

P19. **Hi-Lo with Limited Tries**

Write a program that picks a random number from 1-100. The user keeps guessing as long as their guess is wrong, **and** they've guessed less than 7 times. If their guess is higher than the number, say "Too high." If their guess is lower than the number, say "Too low." When they get it right, the game stops. Or, if they hit seven guesses, the game stops even if they never got it right.
This means your while loop will have a compound condition using &&.



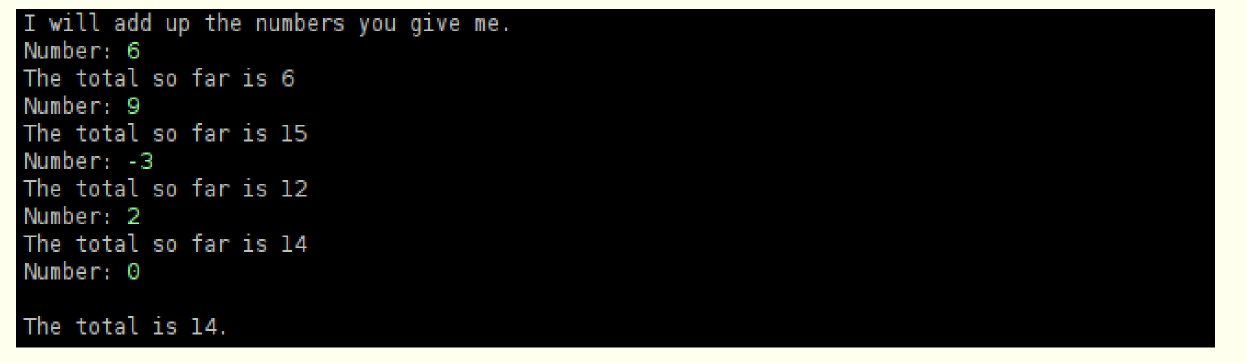
```
I'm thinking of a number between 1-100.  You have 7 guesses.
First guess: 50
Sorry, you are too low.
Guess # 2: 75
Sorry, you are too low.
Guess # 3: 87
Sorry, that guess is too high.
Guess # 4: 82
Sorry, you are too low.
Guess # 5: 84
You guessed it!  What are the odds?!?
```



```
I'm thinking of a number between 1-100.  You have 7 guesses.
First guess: 1
Sorry, you are too low.
Guess # 2: 2
Sorry, you are too low.
Guess # 3: -8
Sorry, you are too low.
Guess # 4: 0
Sorry, you are too low.
Guess # 5: 7
Sorry, you are too low.
Guess # 6: 612
Sorry, that guess is too high.
Guess # 7: -523
Sorry, you didn't guess it in 7 tries.  You lose.
```

P20. **Adding Values in a Loop**

Write a program that gets several integers from the user. Sum up all the integers they give you. Stop looping when they enter a 0. Display the total at the end.
You must use a while loop.



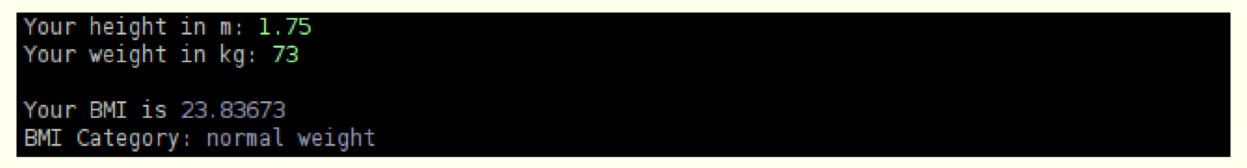
```
I will add up the numbers you give me.
Number: 6
The total so far is 6
Number: 9
The total so far is 15
Number: -3
The total so far is 12
Number: 2
The total so far is 14
Number: 0

The total is 14.
```

P21. **BMI Categories**

Reuse the BMI calculator you wrote previously . Then use some if statements to show the category for a given BMI.

BMI	**category**
less than 18.5	underweight
18.5 to 24.9	normal weight
25.0 to 29.9	overweight
30.0 or more	obese

Note: Although BMI is a very good estimate of human body fat, the formula doesn't work well for athletes with a lot of muscle, or people who are extremely short or very tall. If you are concerned about your BMI, check with your doctor.



```
Your height in m: 1.75
Your weight in kg: 73

Your BMI is 23.83673
BMI Category: normal weight
```

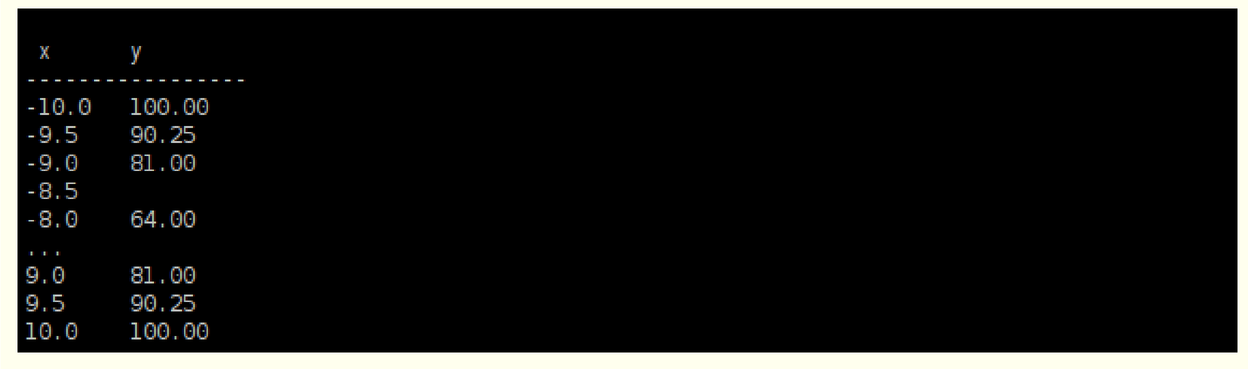
P22. **Xs and Ys**

Write another program that uses a for loop. With the loop, make the variable x go from -10 to 10, counting by 0.5. (This means that x can't be an int.)

Inside the body of the loop, make another variable y become the current value of x squared.

Then display the current values of both x and y.

To get your output to line up like mine, use a tab.



```
 x      y
----------------
-10.0   100.00
-9.5    90.25
-9.0    81.00
-8.5
-8.0    64.00
...
9.0     81.00
9.5     90.25
10.0    100.00
```

P23. Fizz Buzz

Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

P24. Write a program with

1. a function to find area of a triangle using Heron’s formula.
2. Function to find area of a Square
3. Function to find area of a rectangle
4. Function to find area of a Circle.
5. Function to find the area of a trapezium.

P.25 Month Name

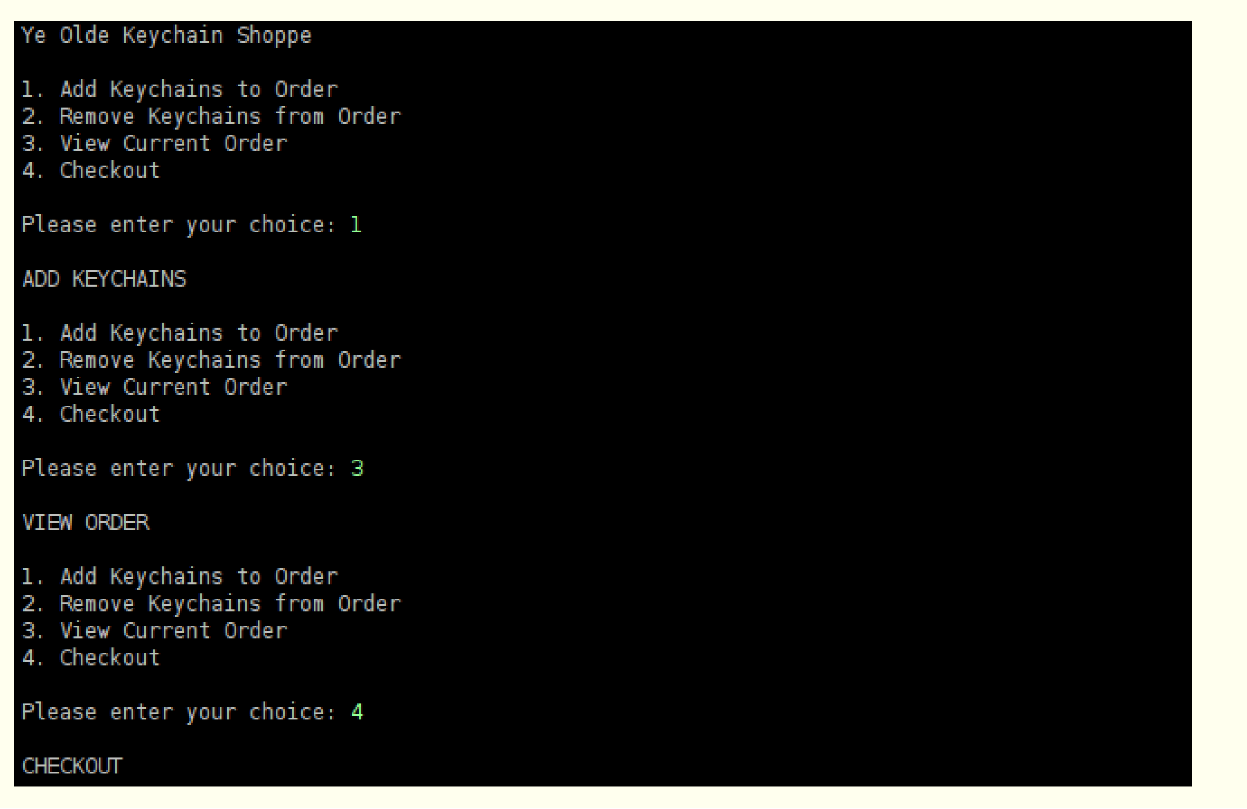
Write a function. It will return the name of a month of the year, given the month number, according to the table below. Make sure you **do not** put any input or output statements in the function; the month number will be *passed in* and the string containing the name will be *returned*.

Number	Month
1	January
2	February
3	March
4	April
5	May
6	June
7	July

8	August
9	September
10	October
11	November
12	December
anything else	error

The function **must** be called month_name(), and must have one parameter (an integer), and return a String.

P26. Write a program that pulls up a menu with 4 options. It should look something like…



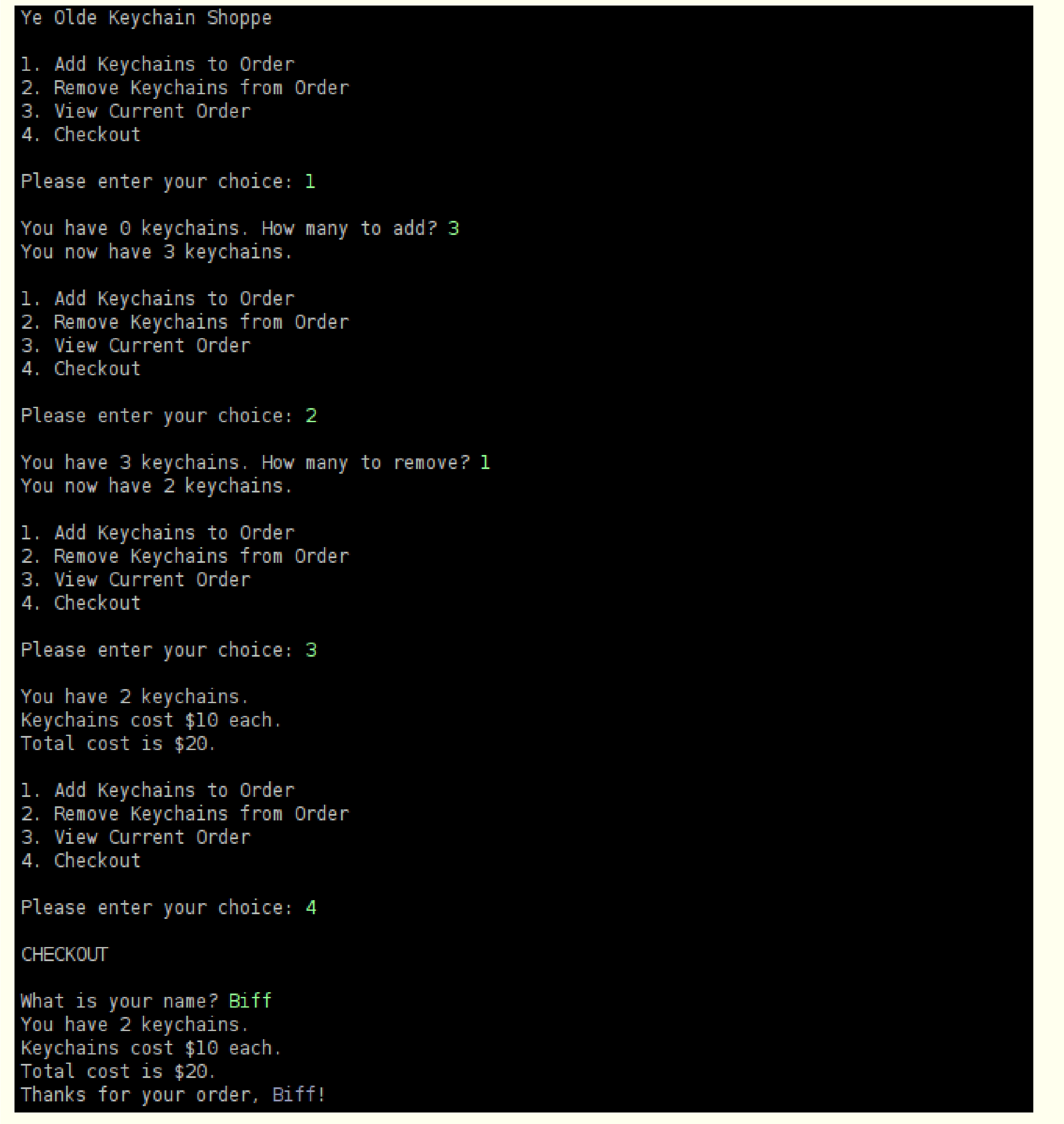
```
Ye Olde Keychain Shoppe

1. Add Keychains to Order
2. Remove Keychains from Order
3. View Current Order
4. Checkout

Please enter your choice: 1

ADD KEYCHAINS

1. Add Keychains to Order
2. Remove Keychains from Order
3. View Current Order
4. Checkout

Please enter your choice: 3

VIEW ORDER

1. Add Keychains to Order
2. Remove Keychains from Order
3. View Current Order
4. Checkout

Please enter your choice: 4

CHECKOUT
```

- You will need to create functions for each of the 4 menu options. Entering the number will call the correct function.
- This assignment does not require you to complete ANY of the functionality except for the working menu system. There is no need for you to program the ability to add keychains, remove keychains, view orders or checkout.
- The functions should be named **add_keychains(), remove_keychains(), view_order() and checkout().**

- Each function should print a message that it has been called.
- The user should be able to keep putting in choices until the **checkout()** function is called. When **checkout()** is finished, the program should end.

P.27 **Keychains for Sale, for real this time**

Okay, now it is time to make the keychain shop actually work.



```
Ye Olde Keychain Shoppe

1. Add Keychains to Order
2. Remove Keychains from Order
3. View Current Order
4. Checkout

Please enter your choice: 1

You have 0 keychains. How many to add? 3
You now have 3 keychains.

1. Add Keychains to Order
2. Remove Keychains from Order
3. View Current Order
4. Checkout

Please enter your choice: 2

You have 3 keychains. How many to remove? 1
You now have 2 keychains.

1. Add Keychains to Order
2. Remove Keychains from Order
3. View Current Order
4. Checkout

Please enter your choice: 3

You have 2 keychains.
Keychains cost $10 each.
Total cost is $20.

1. Add Keychains to Order
2. Remove Keychains from Order
3. View Current Order
4. Checkout

Please enter your choice: 4

CHECKOUT

What is your name? Biff
You have 2 keychains.
Keychains cost $10 each.
Total cost is $20.
Thanks for your order, Biff!
```

- You will need 2 new variables , one to store the current number of keychains, and one to store the price per keychain.

- The price should be $10 per keychain.
- **add_keychains()** will need to be passed one int, and have to return an integer. It will ask the user for the number of keychains to add to the order, and then return the new number of keychains.
- **remove_keychains()** will need to be passed one int, and have to return an integer. It will ask the user for the number of keychains to remove from the order, and then return the new number of keychains.
- **view_order()** will need to be passed two ints. It will display, on three different lines, the number of keychains in the order, the price per keychain, and the total cost of the order.
- **checkout()** will need to be passed two ints. It will ask the user for his/her name in order to deliver them correctly, display the order information, and then thank the user, by name, for ordering.

P.28 Keychains for Sale, real ultimate power

You're going to add some error checking and additional features, to Keychains2.
You need to make sure that the user always has a positive number, or 0, of keychains in the order.
You need to check for a valid menu choice. If not, display an error message and show the menu again.
You will need 3 new variables in main, one to store the sales tax (8.25%), one to store the shipping cost per order ($5.00), and one to store the additional per keychain shipping cost ($1.00).
view_order() will need to be passed the three additional variables, a total of five.. It will display, on different lines, the number of keychains in the order, the price per keychain, the shipping charges on the order, the subtotal before tax, the tax on the order, and the final cost of the order.
view_order() might look like **view_order(num_keychains, price_per_keychain, tax, base_shipping, per_keychain_shipping)**
checkout() will need to be passed the same values as **view_order()**. It will ask the user for his/her name in order to deliver them correctly, then call **view_order()** to display the order information, and then thank the user, by name, for ordering.

P.29 Please design a command based Calculator application using OOPs whereever necessary. Operations should include add, sub, multiply, divide, square, sq. root. Also, enable your calculator to execute an expression, say (2x5)/7+6x9 using BODMOS rule of precedence.