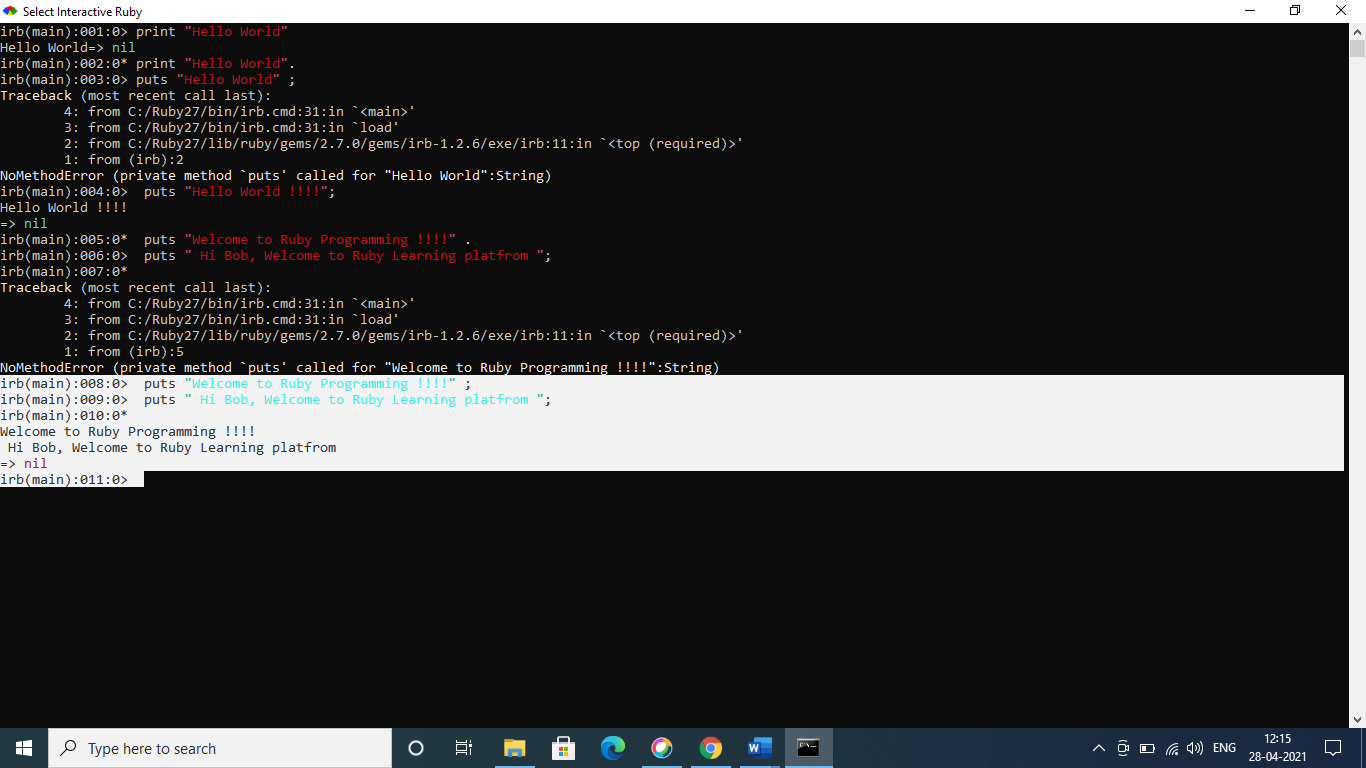
**Sample Ruby Program**

**Q.1 Print a sample welcome message to greet the user**

**Q.2 Get the user’s name at run time and greet them**

**Output**



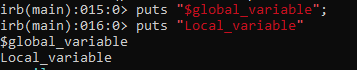
**Q.3) Working with numbers – Create two variables (float) perform multiplication and display the output**

**Output:**



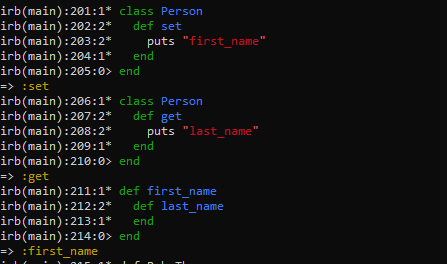
**Q.4) Local variables and global variables**

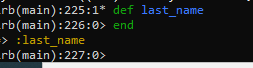
* 1. Create two variables one is local variable and another is global variable( starts with $)
  2. Create two methods and resign the local variables and print local variable and global variable from both the methods.



**Q.5. Working with classes**

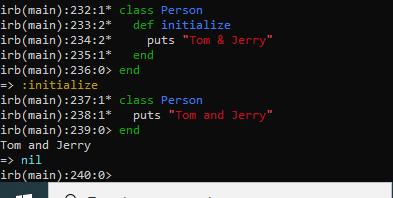
* 1. Create a class called person,Declare the setter and getter method for the name attribute
  2. Declare a method full\_name to return first\_name + last\_name





Q.6 **Working with classes and constructors**

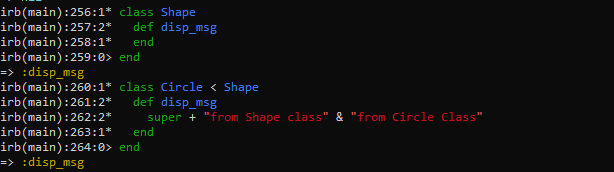
* 1. Use the same person class as above
  2. When creating a new object, instantiate with both first name and last name(Hint: use **initialize**)
  3. Declare a method full\_name to return first\_name + last\_name

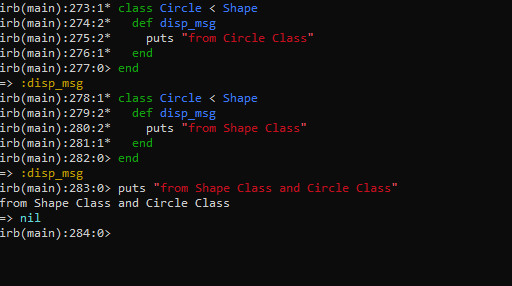


Q.7 **Classes and inheritance**

* 1. Create class called Circle and inherit it from the Shape class by creating shape class
  2. Create a method disp\_msg in Shape class and just print “from shape class”
  3. Create a method disp\_msg in Circle class
  4. Now inside the disp\_msg from Circle class you need to print two messages “from shape class” and “from Circle Class”. Hint Use **Super**

**Output:**





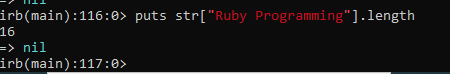
**Q.8) Working with strings**

* 1. Create a string “Ruby Programming”

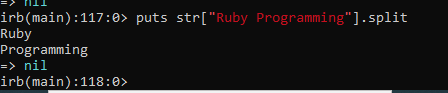
v

b) Print out the total length of the string

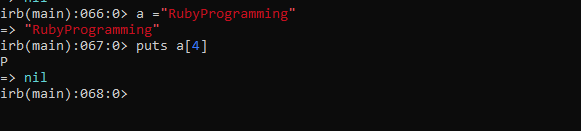
output:



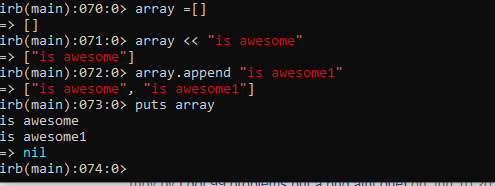
c) Split the string one character each



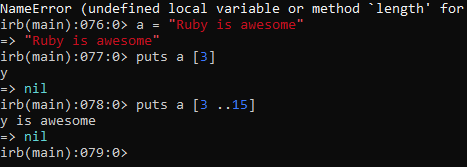
d.Print the value of string of the 4th element



e.Append “is awesome!” to the existing string

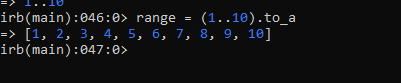


f. Print all the letters starting from the 3rd position

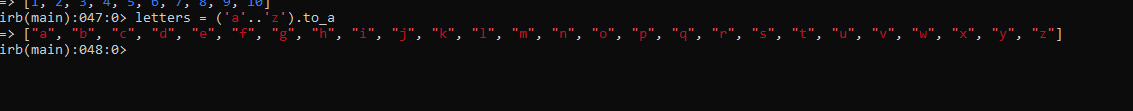


**Q.9 Working with ranges**

* 1. Create a range of numbers between 1 and 10

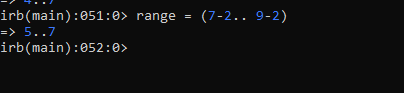


* 1. Create a range of alphabets from ‘z’ to ‘a’



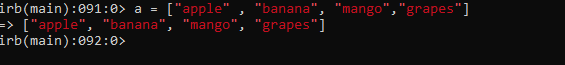
* 1. After creating number ranges print the value of each item multiplied by 2 if it is even number and if it is odd subtract 2 and print the entire results.



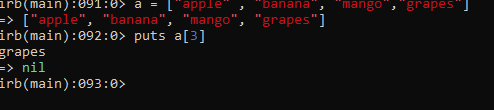


Q.10 **Working with arrays**

* 1. **Create an array with fruit names**



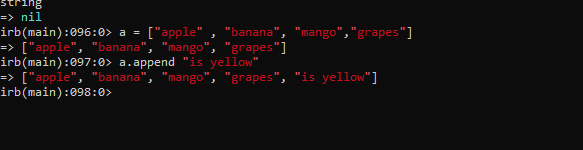
* 1. Print the fruit name at position 3



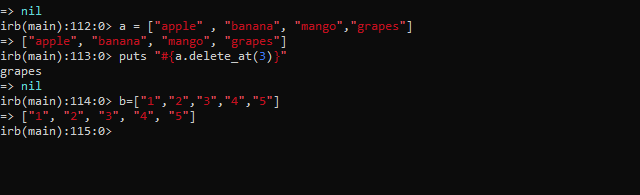
* 1. Print each item from array



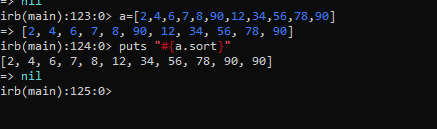
* 1. Append “is yellow” if the fruit name is mango



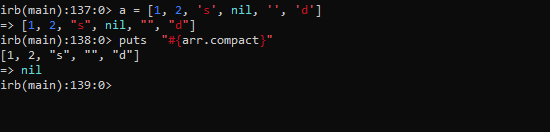
* 1. Delete 3 item from the array and print the new array



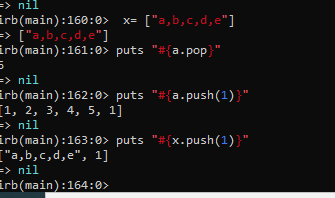
* 1. Sorting an array in ascending order



* 1. Assign nill value to the array and remove that nil value from array

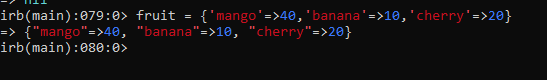


* 1. Create array inside an array and remove the array inside the outer array example
     1. [a, b, [c, d], e] and I want the output like [a, b, c, d, e]

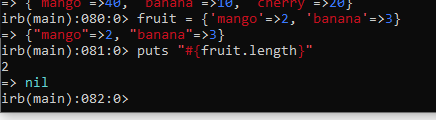


**Q.11 Working with hashes**

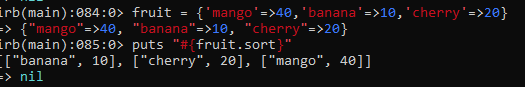
* 1. Create a hash with some values



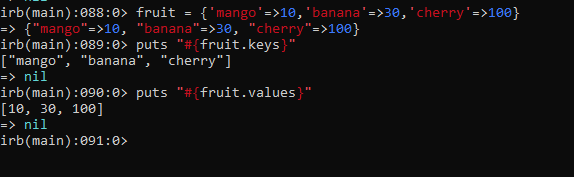
* 1. Print the length of the hash



* 1. Sort the hash ascending order

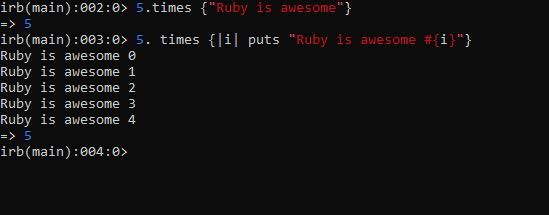


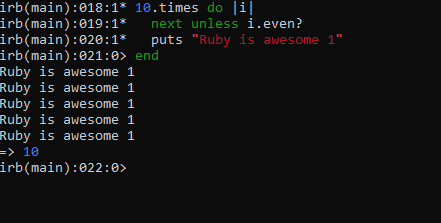
d.Print the keys of hash & e. Print the values of the hash



**Q. 12. Loops**

* 1. Write a loop to print “Ruby is awesome” 5 times



b.Same as above, but you need to add the iteration to the result “Ruby is awesome 1” .

c.Create a loop so that it prints the values in descending order 10, 9 , 8 7 and so on Hint use **downTo.**

