EC606PC: SCRIPTING LANGUAGES LAB MANUAL

Prepared by:

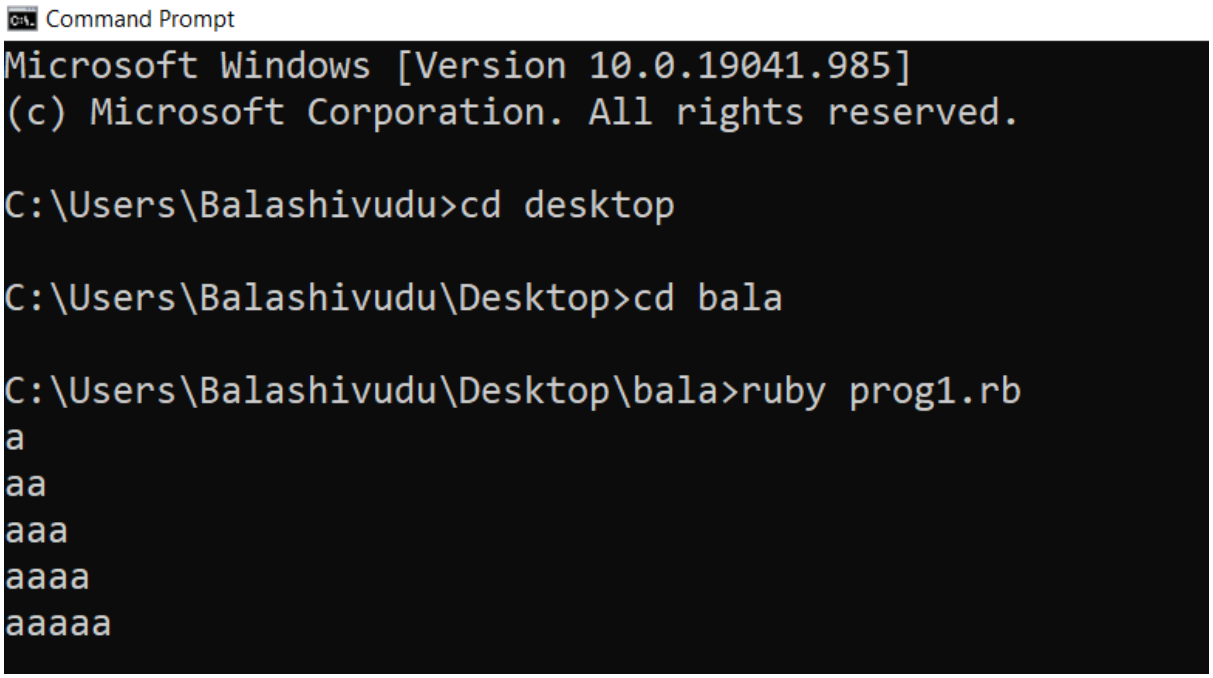
Dr. M. Shoban, Associate Professor, NRCM.

#1 Write a Ruby program to create a new string which is n copies of a given string where n is a non-negative integer.

Ruby Code:

```
def multiple_string(str, n)
    return str*nend
print multiple_string('a', 1),"\n"
print multiple_string('a', 2),"\n"
print multiple_string('a', 3),"\n"
print multiple_string('a', 4),"\n"
print multiple_string('a', 5),"\n"
```

Output:

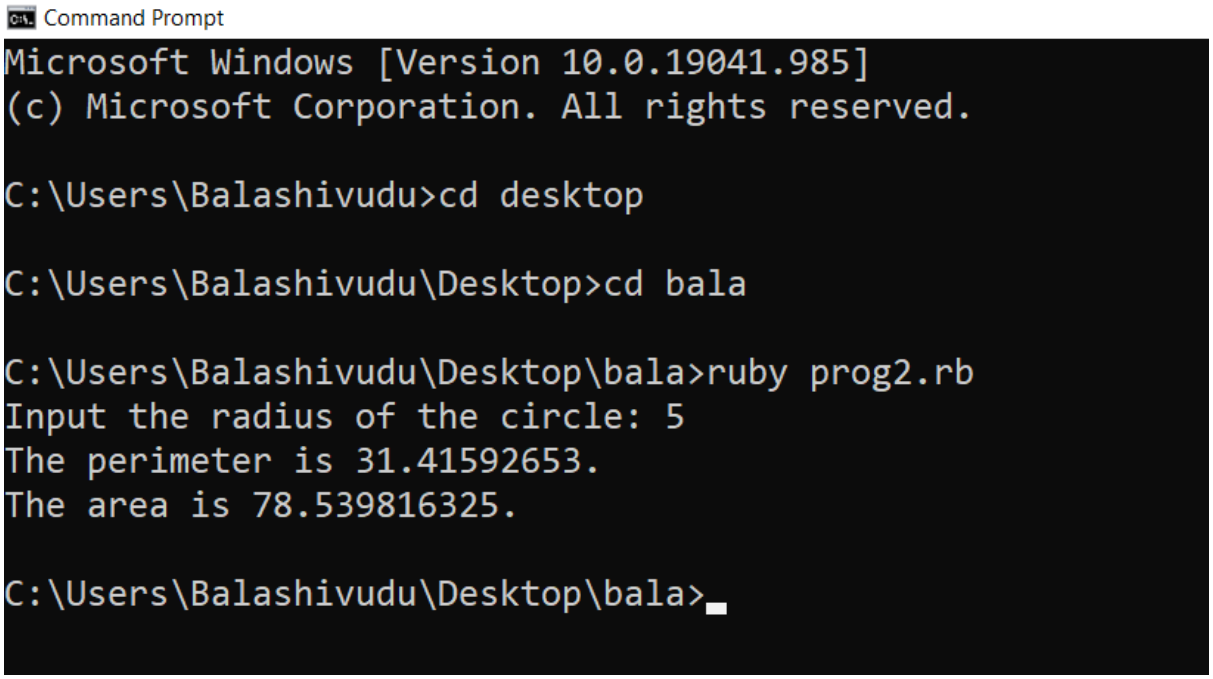


#2 Write a Ruby script which accepts the radius of a circle from the user and compute the perimeter and area.

Ruby Code:

```
radius = 5.0
perimeter = 0.0
area = 0.0
print "Input the radius of the circle: "
radius = gets.to_f
perimeter = 2 * 3.141592653 * radius
area = 3.141592653 * radius * radius
puts "The perimeter is #{perimeter}."
puts "The area is #{area}."
```

Output:



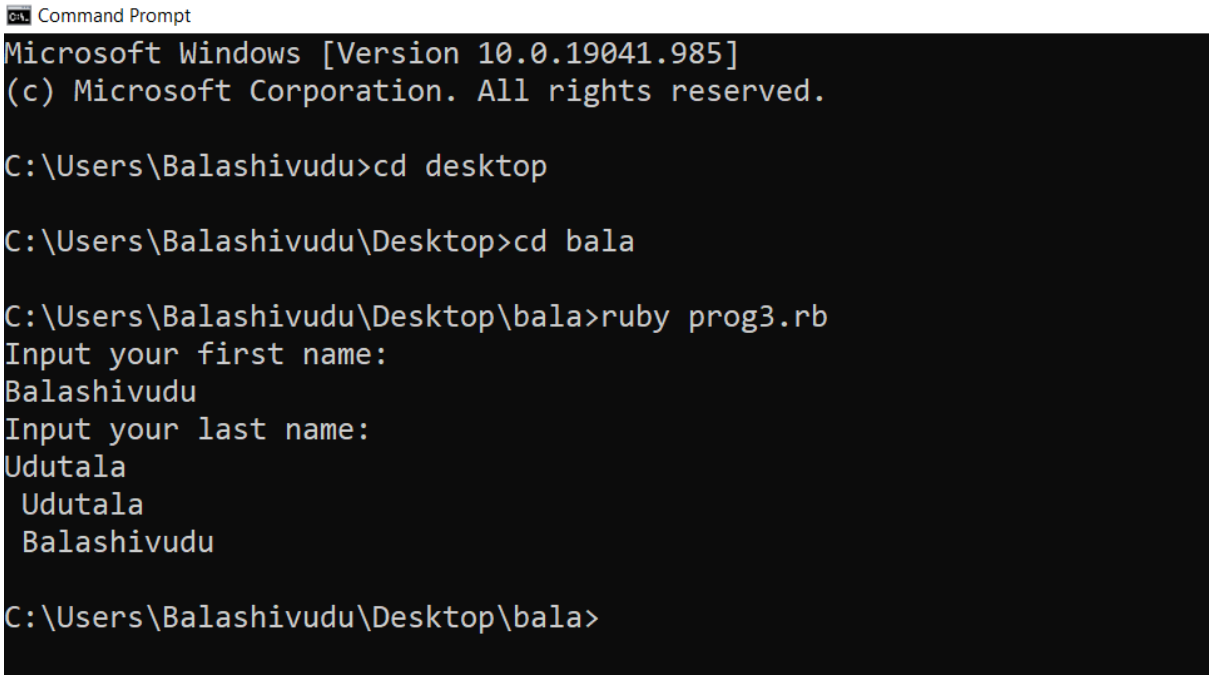
#3. Write a Ruby program which accept the user's first and last name and print them in reverse order with a space between them.

Ruby Code:

```
puts "Input your first name: "
fname = gets
puts "Input your last name: "
lname = gets
```

puts " #{lname} #{fname}"

Output:



#4. Write a Ruby script to accept a filename from the user print the extension of that

```
file = "/user/system/test.rb"     # file name
fbname = File.basename file
puts "File name: "+fbname    # basename
bname = File.basename file,".rb"
puts "Base name: "+bname     # file extention
ffextn = File.extname  file
puts "Extention: "+ffextn     # path name
path_name= File.dirname  file
puts "Path name: "+path_name
```

Output:

```
C:\Users\Balashivudu\Desktop\bala>ruby prog4.rb
File name: test.rb
Base name: test
Extention: .rb
Path name: /user/system

C:\Users\Balashivudu\Desktop\bala>_
```

 # 5. Write a Ruby program to find the greatest of three numbers

Ruby Code:

```
x,y,z = 2,5,4

if x >= y and x >= z
    puts "x = #{x} is greatest."
elsif y >= z and y >= x
    puts "y = #{y} is greatest."
else
    puts "z = #{z} is greatest."
End
```

Output:

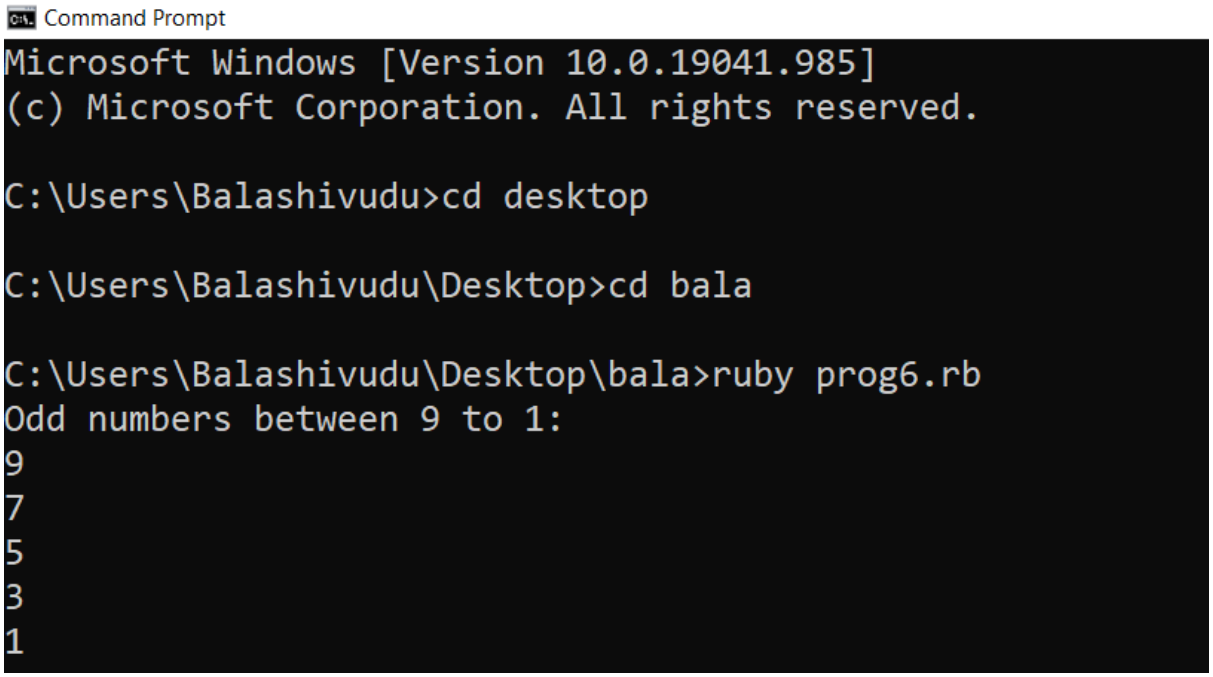
```
C:\Users\Balashivudu\Desktop\bala>ruby prog5.rb
y = 5 is greatest.

C:\Users\Balashivudu\Desktop\bala>
```

#6. Write a Ruby script to print odd numbers from 10 to 1

Ruby Code:

```
puts "Odd numbers between 9 to 1: "
9. step 1, -2 do |x|
    puts "#{x}"
end
```

Output:

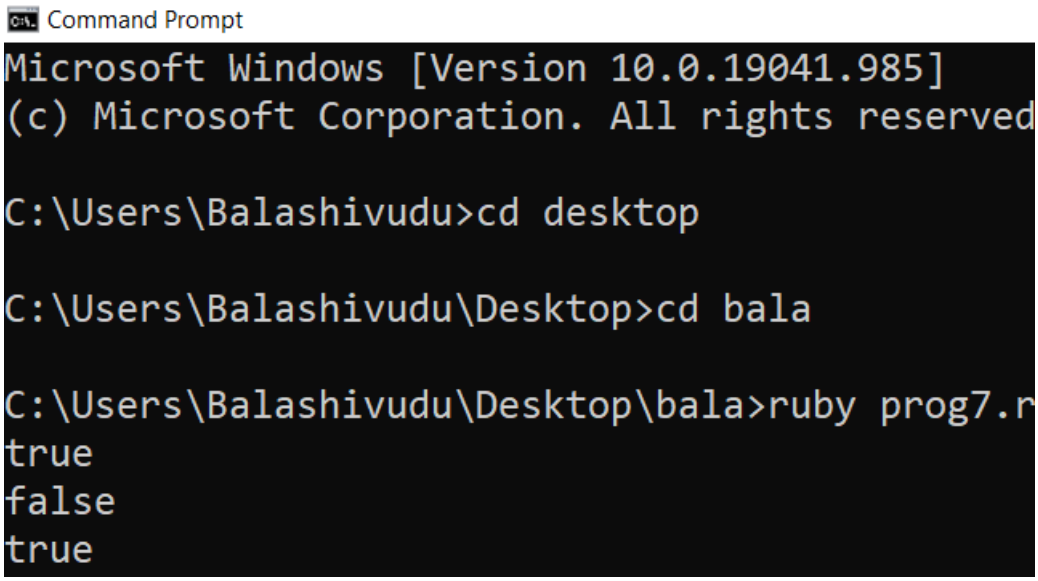


#7. Write a Ruby scirpt to check two integers and return true if one of them is 20 or #their sum is 20 or else return false

Ruby Code:

```
def makes20(x,y)
   return x == 20 || y == 20 || x + y == 20
end

print makes20(10, 10),"\n"
print makes20(40, 10),"\n"
print makes20(15, 20)
```

Output:

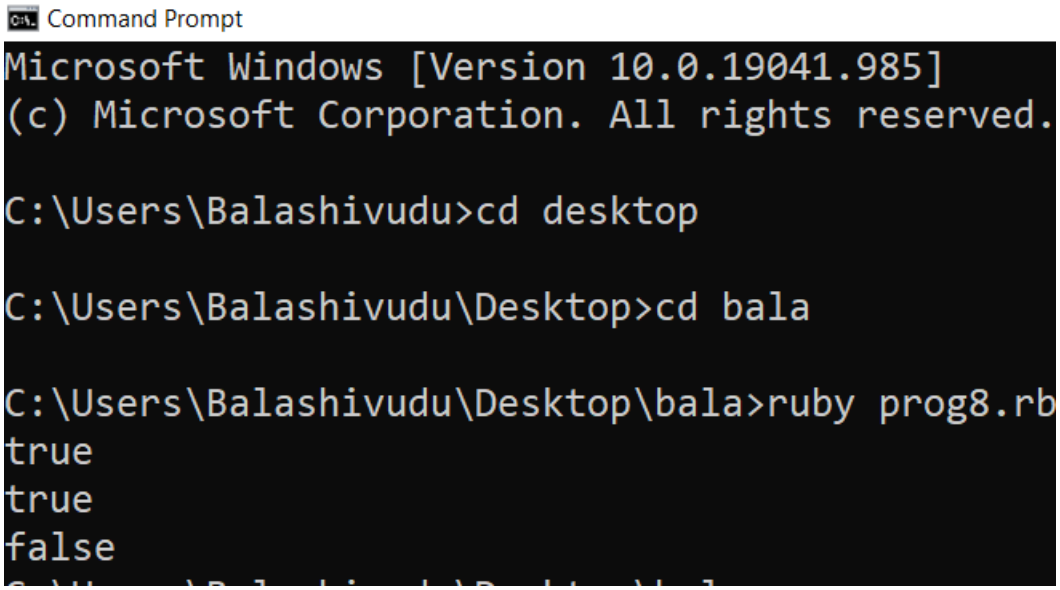


#8. Write a Ruby program to check two temperatures and return true if one is less than 0 and the other is greater than 100.

Ruby Code:

```
def temp(temp1, temp2)
   return ( temp1 < 0 && temp2 > 100 ) || ( temp1 > 100 && temp2 < 0 );
end
print temp(110, -1),"\n"
print temp(-1, 110),"\n"
print temp(2, 120)
```

Output:

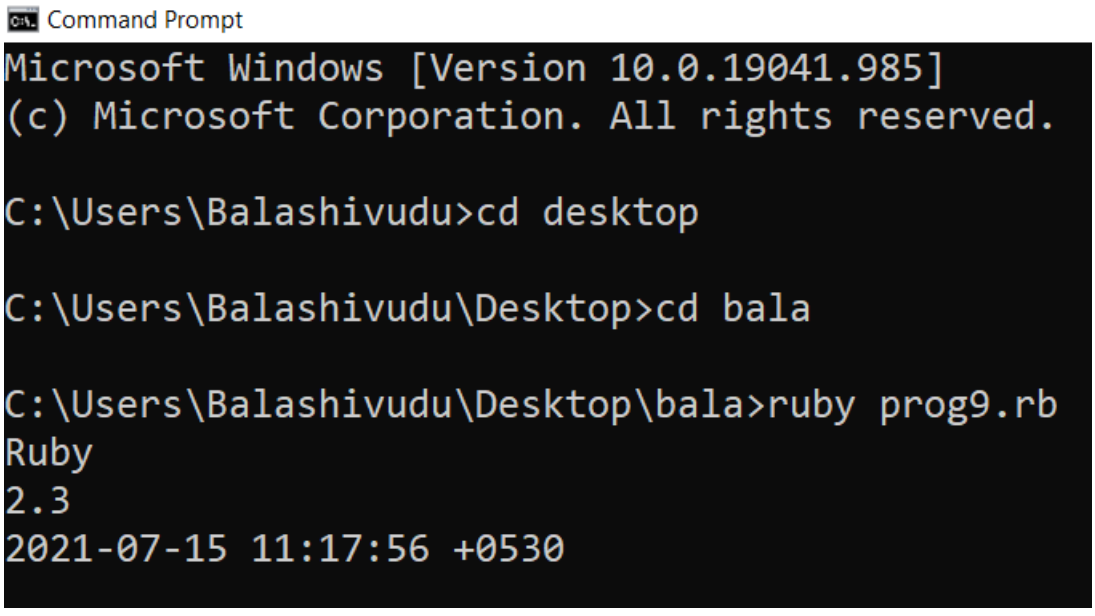


#9 Write a Ruby script to print the elements of a given array

Ruby Code:

```
array1 = ["Ruby", 2.3, Time.now]
for array_element in array1
    puts array_element
end
```

Output:



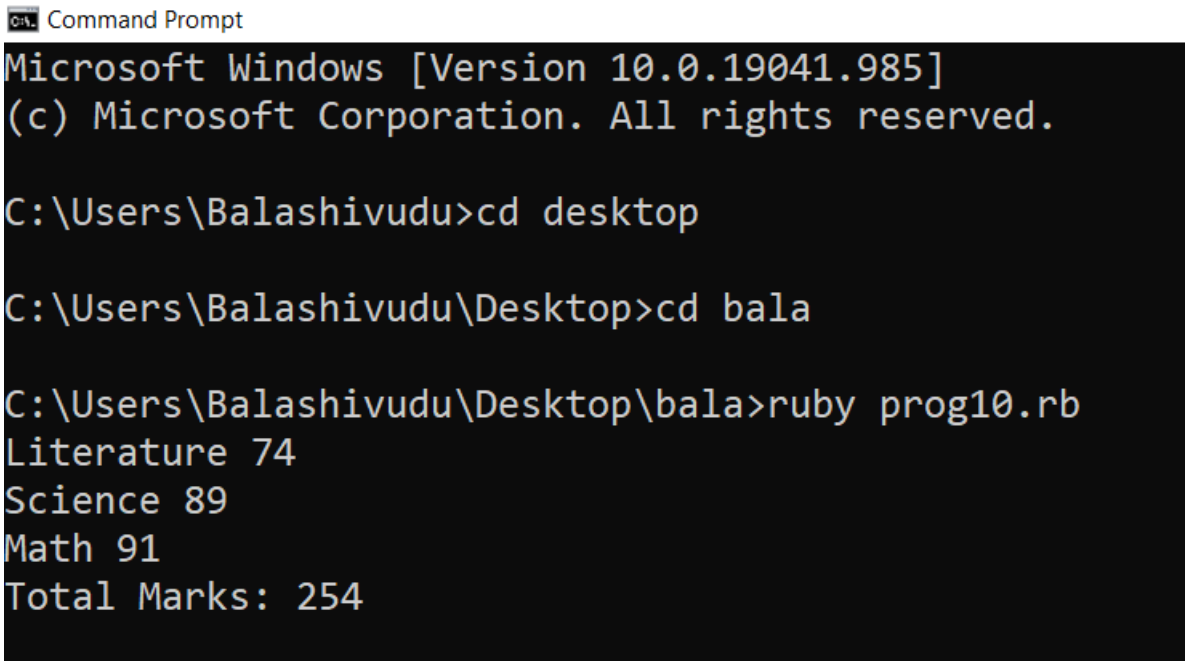
#10. Write a Ruby program to retrieve the total marks where subject name and marks of a student stored in a hash.

#Sample subject and marks : Literature -74, Science – 89, Math-91

Ruby Code:

```
student_marks  = Hash.new 0

student_marks['Literature']  = 74

student_marks['Science']  = 89

student_marks['Math']  = 91

total_marks = 0

student_marks.each{|key,value|

            puts "#{key} #{value}"

            total_marks +=value

    }
```

Output:



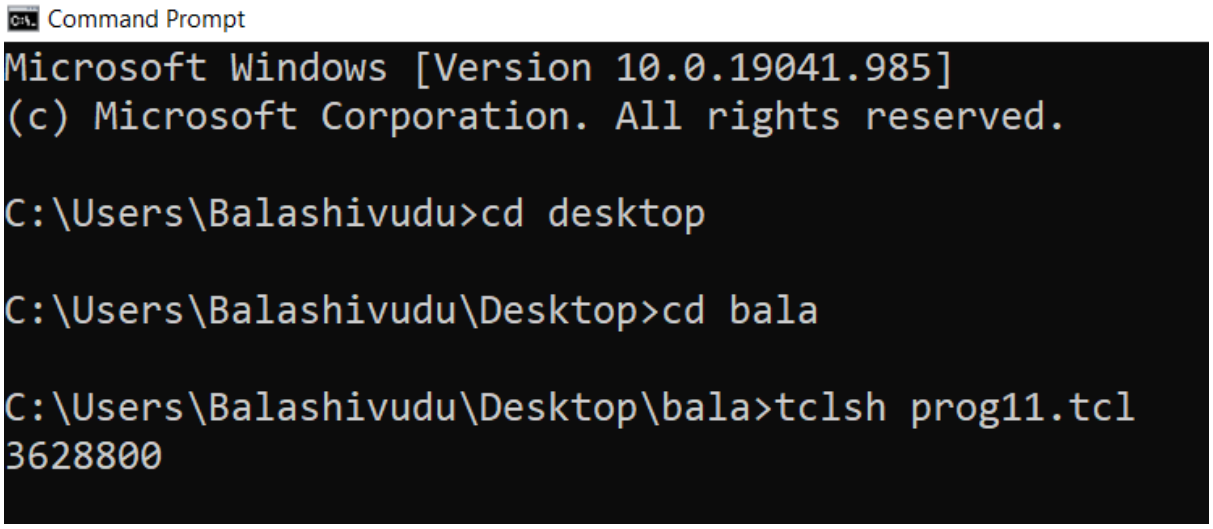
#11 Write a TCL script to find the factorial of a number

TCL Code:

```
proc Factorial {x} {
```

```
  set i 1; set product 1
  while {$i <= $x} {
    set product [expr $product * $i]
    incr i
  }
  return $product
}
puts [Factorial 10]
```

Output:



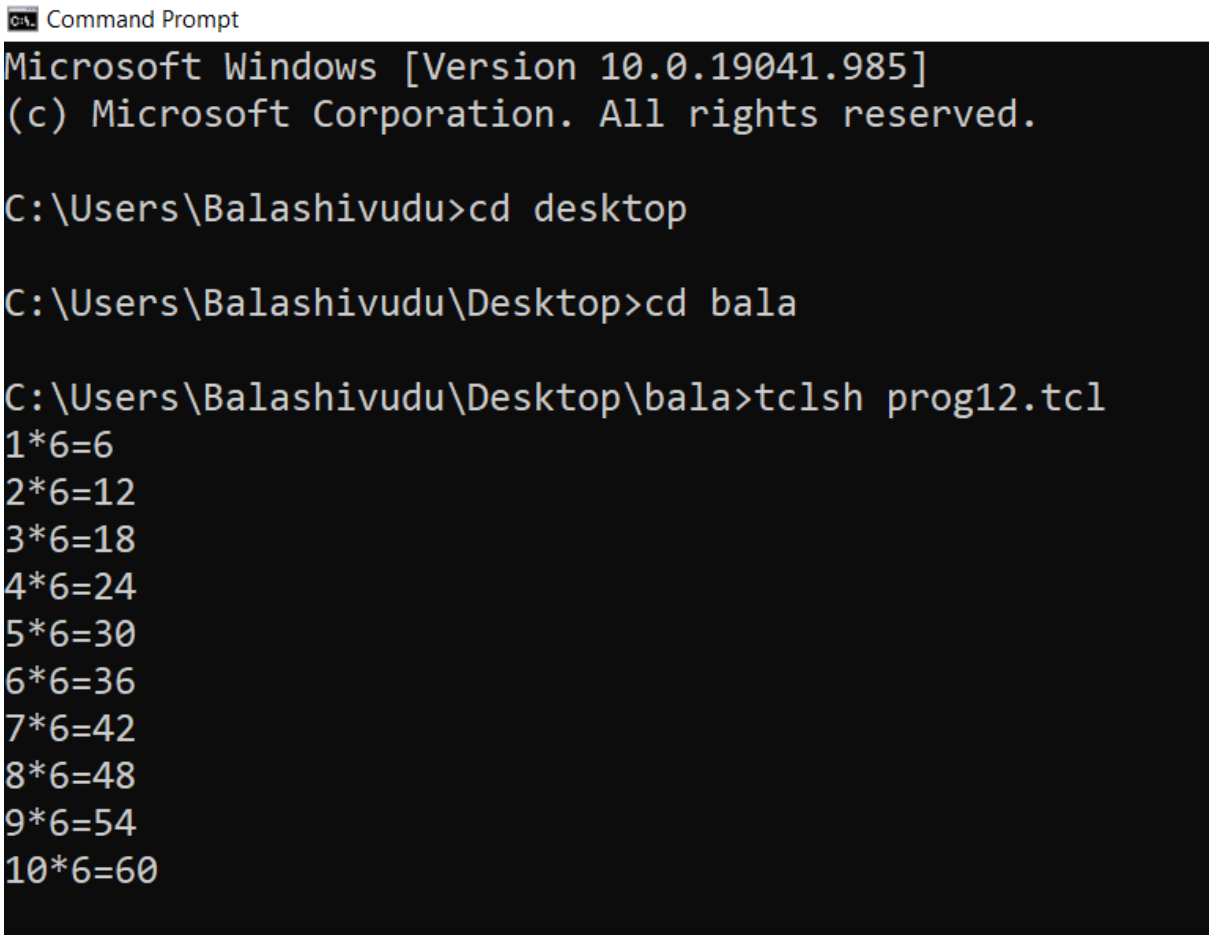
#12 Write a TCL script that multiplies the numbers from 1 to 10]

#!/usr/bin/tclsh

TCL Code:

```
  set v 6
  for {set n 1} {$n < 11} {incr n} {
    set res [expr {$n*$v}]
    puts $n*$v=$res

  }
```

Output:



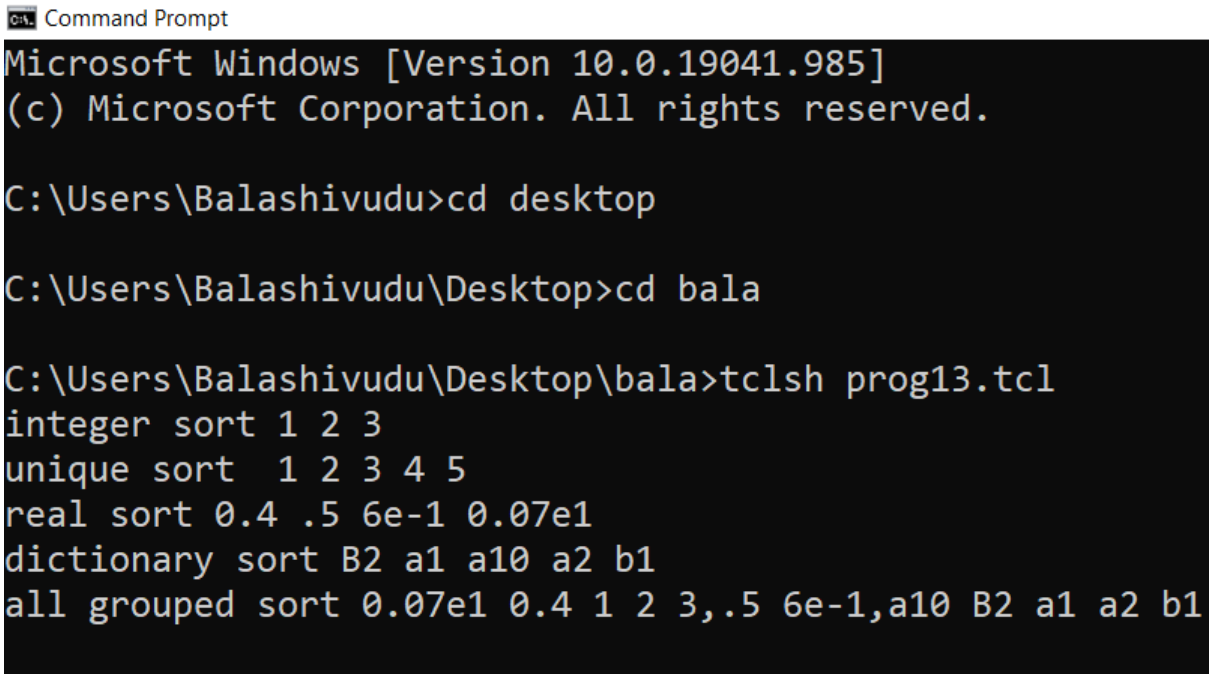
#13.Write a TCL script for Sorting a list using a comparison function

TCL Code:

```
#!/usr/bin/tclsh
set l {2 1 3}
set intset [lsort -integer $l]
puts "integer sort $intset"
set lu {2 1 1 2 4 5 5 3}
set unq [lsort -unique $lu]
puts "unique sort $unq"
set rl {.5 0.07e1 0.4 6e-1}
set slr [lsort -real $rl]
puts "real sort $slr"
```

```
set dic {a10 B2 b1 a1 a2}

set sdic [lsort $dic]

puts "dictionary sort $sdic"

set all [lsort $l,$rl,$dic]

puts  "all grouped sort $all"

puts "Total Marks: "+total_marks.to_s
```

Output:

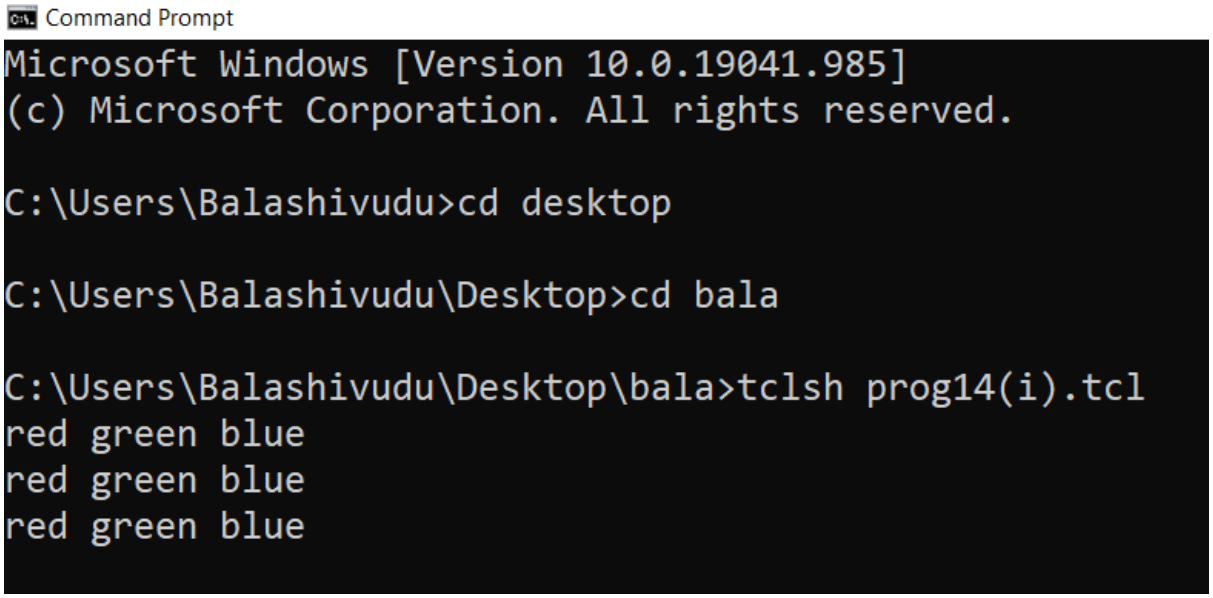


#14.(i)Write a TCL script to create a list

TCL Code:

```
#!/usr/bin/tclsh

set colorList1 {red green blue}; set colorList2 [list red green blue]; set colorList3 [split
"red_green_blue"  _]

puts $colorList1

puts $colorList2

puts $colorList3
```

Output:

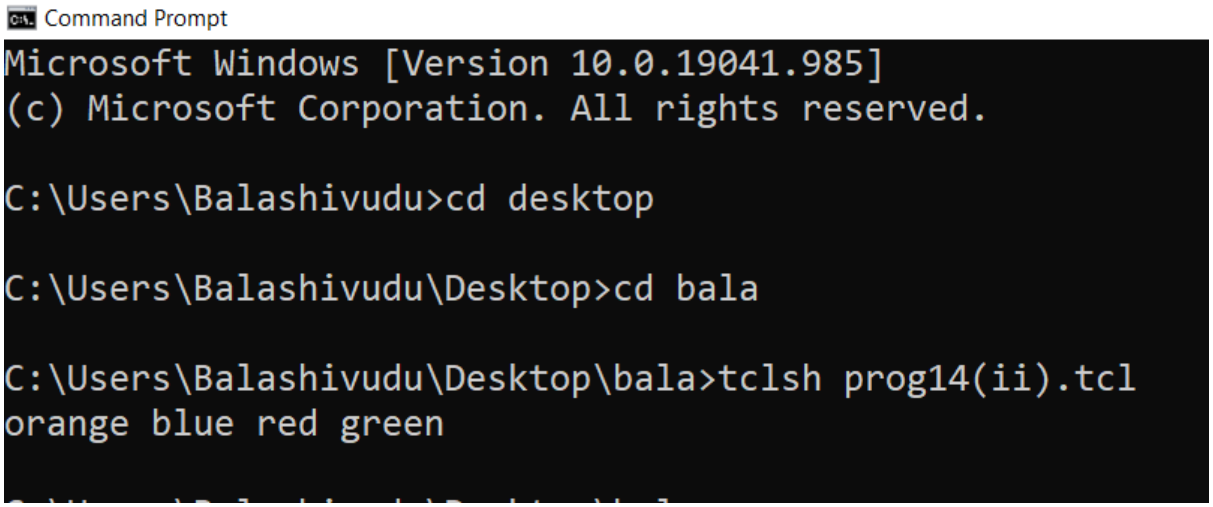


 # 14.(ii) write a TCL script to append elements to the list

TCL Code:

```
#!/usr/bin/tclsh
set var orange
append var " " "blue"
lappend var "red"
lappend var "green"
puts $var
```

Output:

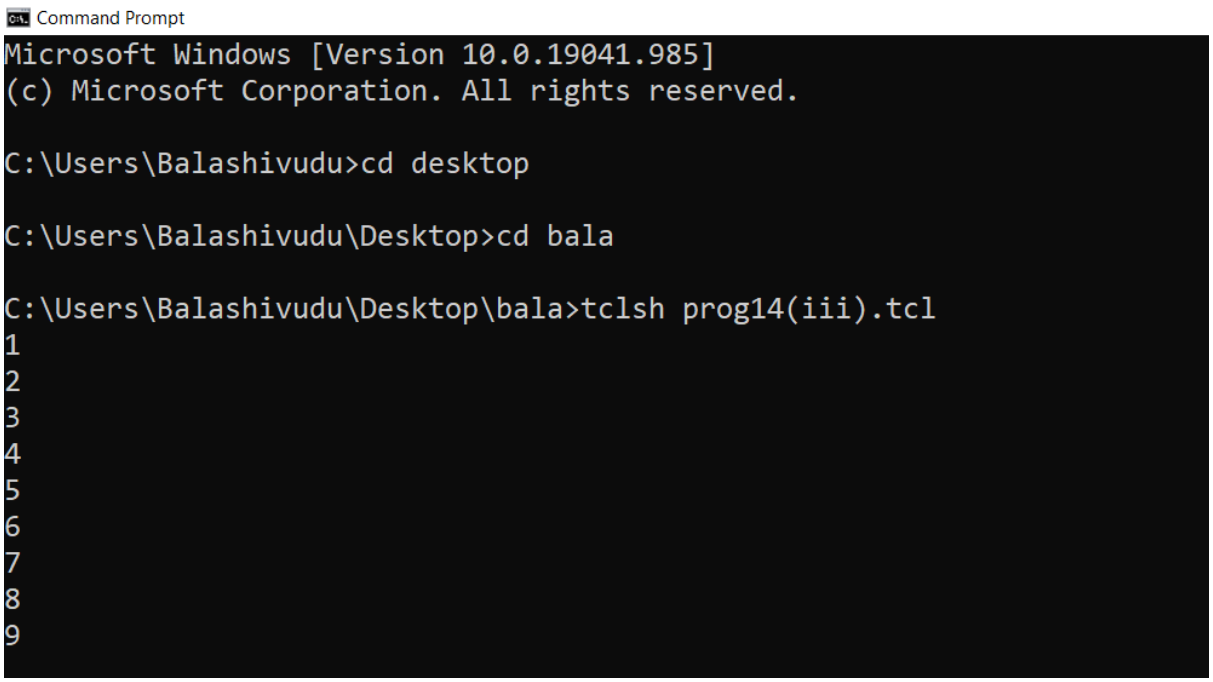


#14.(iii) Write a TCL script to Traverse the list

TCL Code:

```
#!/usr/bin/tclsh
foreach item {1 2 3 4 5 6 7 8 9} {
   puts $item
}
```

Output:



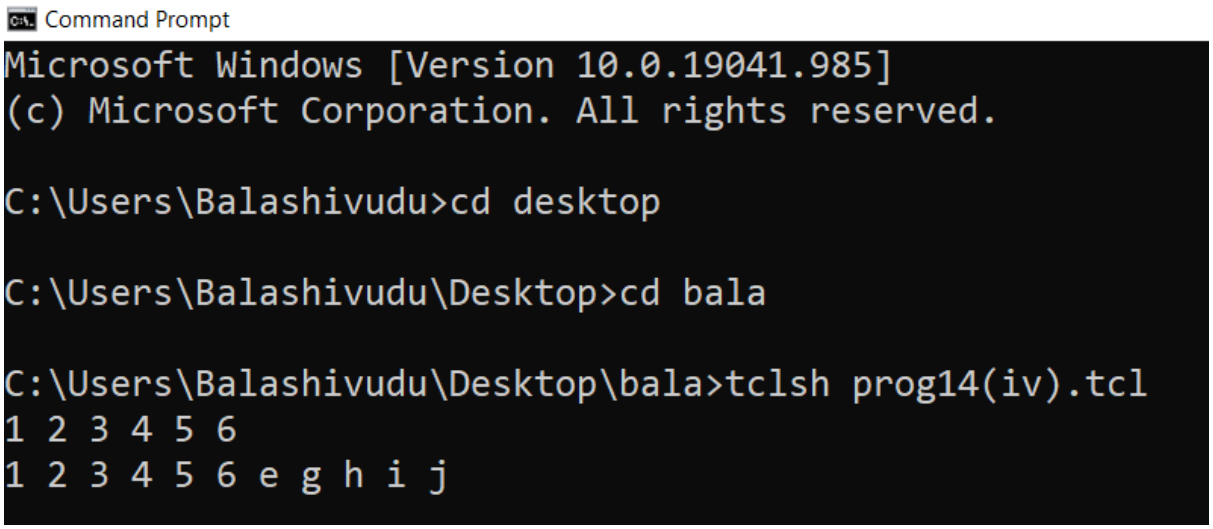
#14.(iv) Write a TCL script to Concatenate the list

TCL Code:

```
#!/usr/bin/tclsh
set A {1 2 3}
set B {4 5 6}
set C {e g h i j }
set x [concat   $A $B ]
set y [concat  $x  $C]
puts  $x
```

```
puts $y
```

Output:



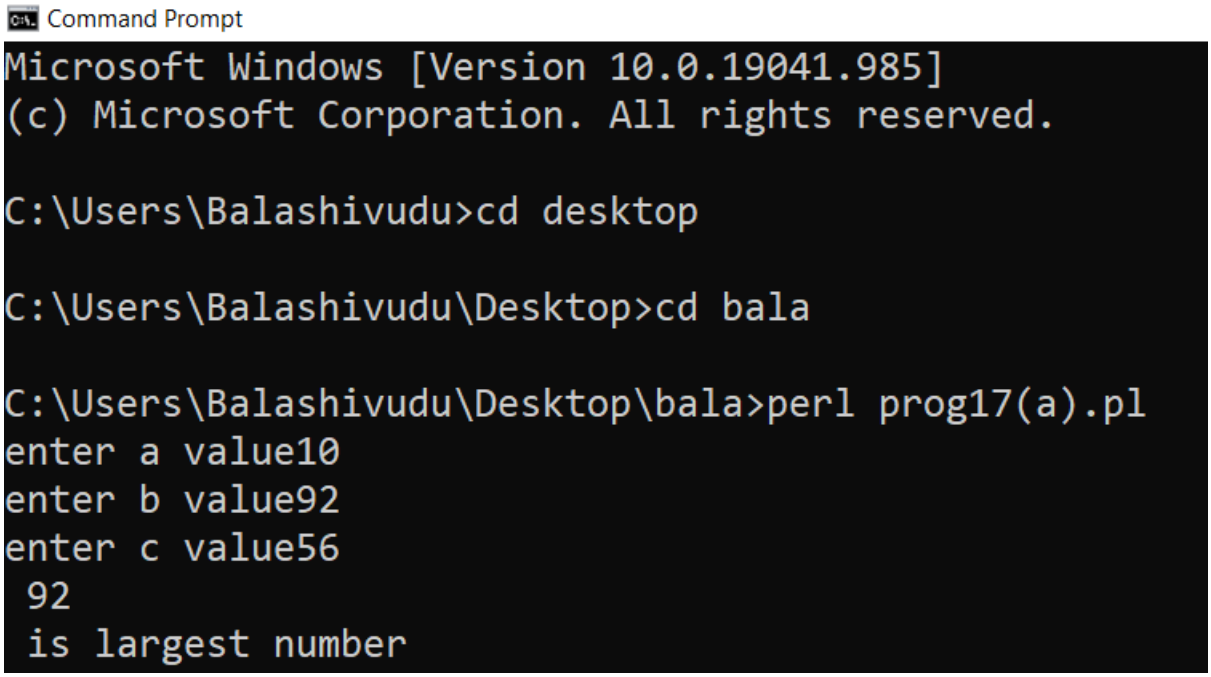
#17.a) Write a Perl script to find the largest number among three numbers

PERL Code:

```
#!/usr/bin/perl
print "enter a value";
$a=<stdin>;
print "enter b value";
$b=<stdin>;
print "enter c value";
$c=<stdin>;
if($a > $b)
{
        if($a> $c)
        {
                print " $a is largest number\n";
        }
        else
        {
```

```
                print " $c is largest number\n";
        }
}
elsif($b >$c)
{
    print " $b is largest number";
}
else
{
   print " $c is largest nnumber";
}
```

Output:



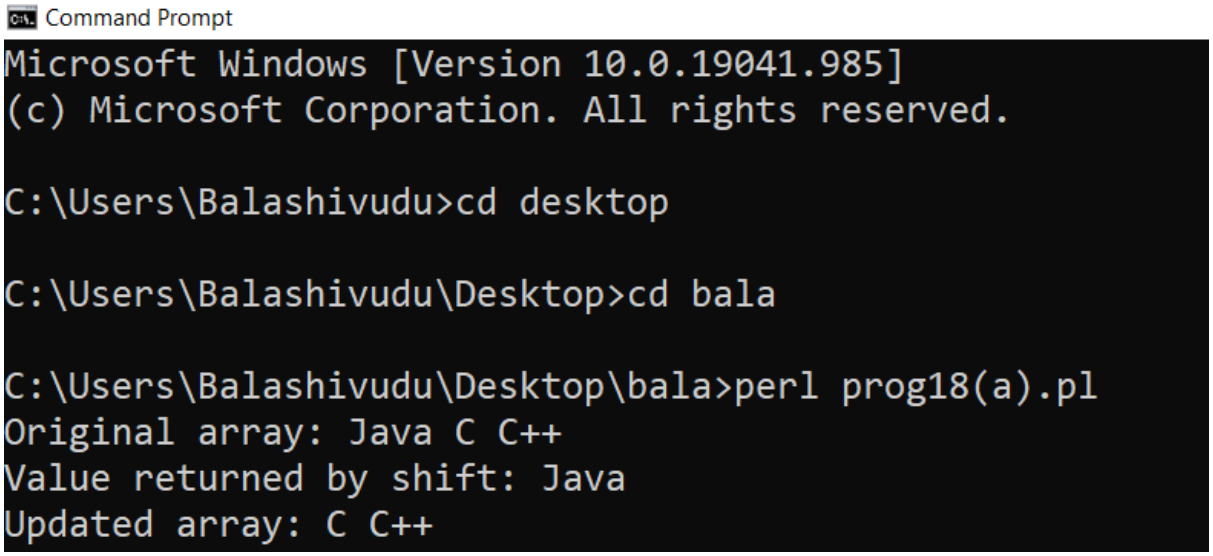
#18. a) Write a Perl program to implement manipulating function shift

PERL Code:

```
#!/usr/bin/perl
# Initalizing the array
@x = ('Java', 'C', 'C++');
```

```
# Print the Inital array
print "Original array: @x \n";

# Prints the value returned
# by shift function
print "Value returned by shift: ",
                shift(@x);

# Array after shift operation
print "\nUpdated array: @x";
```

Output:



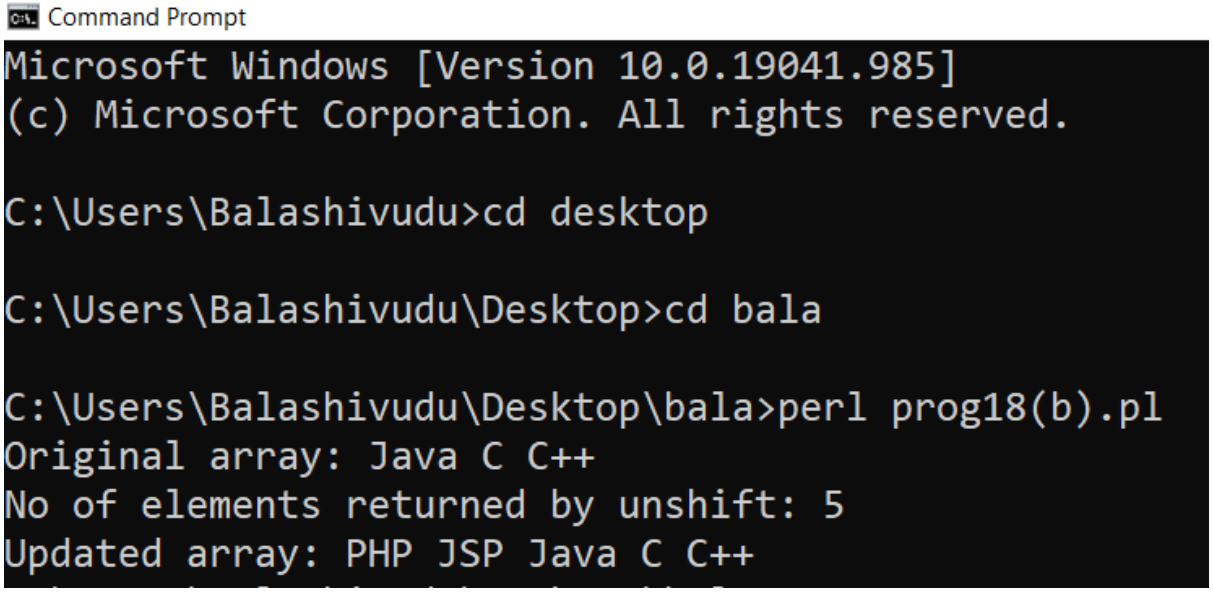
#18. b) Write a Perl program to implement manipulating function unshift

PERL Code:

```
#!/usr/bin/perl
# Initalizing the array
@x = ('Java', 'C', 'C++');

# Print the Inital array
```

```
print "Original array: @x \n";
 # Prints the number of elements
# returned by unshift
print "No of elements returned by unshift: ",
            unshift(@x, 'PHP', 'JSP');

# Array after unshift operation
print "\nUpdated array: @x";
```

Output:



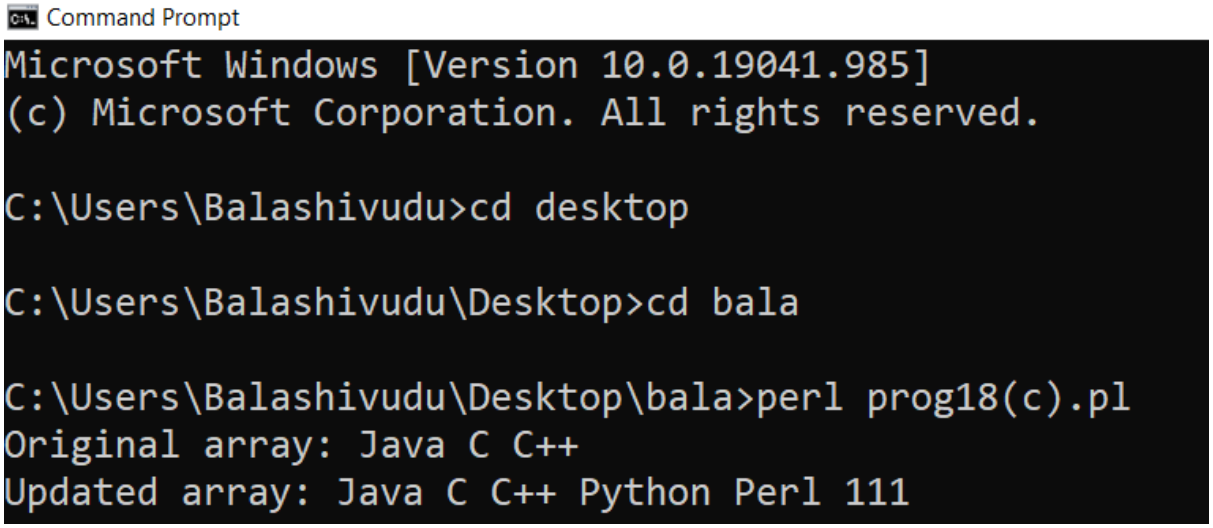
#18. c)Write a Perl program to implement manipulating function Push

PERL Code:

```
#!/usr/bin/perl
 # Initalizing the array
@x = ('Java', 'C', 'C++');

# Print the Inital array
print "Original array: @x \n";
```

```
# Pushing multiple values in the array
push(@x, 'Python', 'Perl',111);

# Printing the array
print "Updated array: @x";
```

Output:

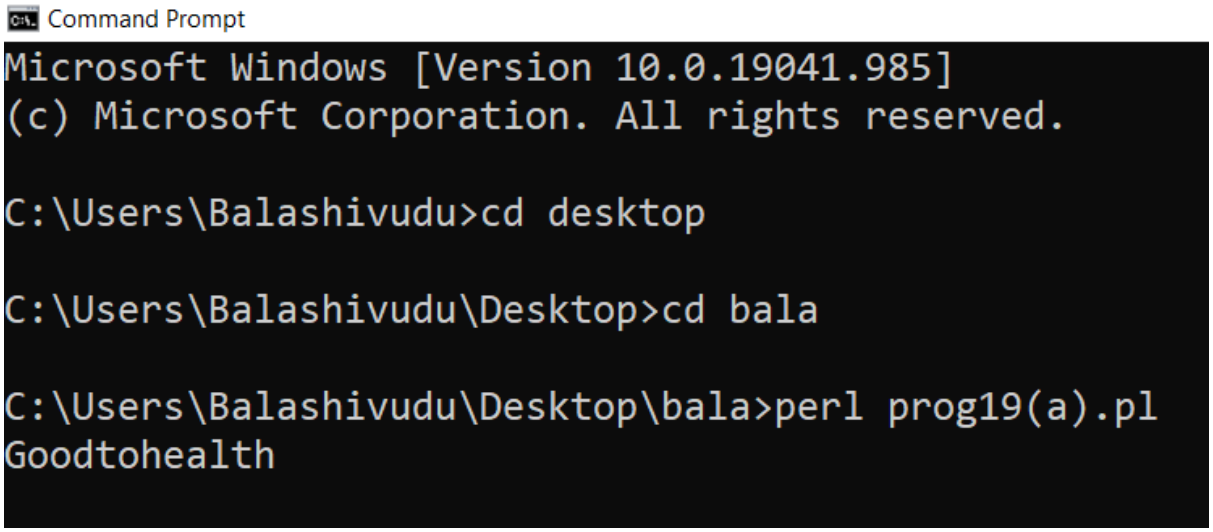


#19. a) Write a Perl script to substitute a word, with another word in a string.

PERL Code:

```
#!/usr/bin/perl -w
# String in which text is to be replaced
$string = "Goodforhealth";

# Use of s operator to replace text with pattern
$string =~ s/for/to/;

# Printing the updated string
print "$string\n";
```

Output:

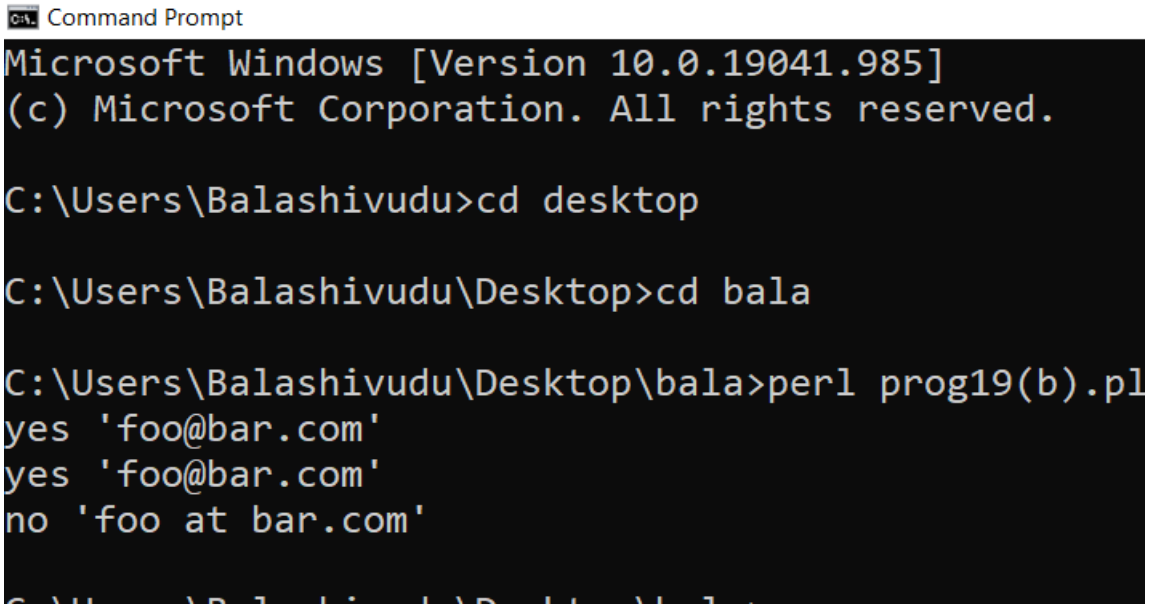


#19.b) Write a Perl script to validate email address.

PERL Code:

```
use strict;
use warnings;
use 5.010;
use Email::Valid;
foreach my $email ('foo@bar.com', ' foo@bar.com ', 'foo at bar.com') {
   my $address = Email::Valid->address($email);
   say ($address ? "yes '$address'" : "no '$email'");
}
```

Output:


Command Prompt
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Balashivudu>cd desktop
C:\Users\Balashivudu\Desktop>cd bala
C:\Users\Balashivudu\Desktop\bala>perl prog19(b).pl
yes 'foo@bar.com'
yes 'foo@bar.com'
no 'foo at bar.com'