

```

# Функція для визначення пріоритету оператора
def пріоритет(оператор)
  case оператор
  when '+', '-' then 1
  when '*', '/' then 2
  else 0
  end
end

# Функція для перетворення інфіксного виразу у зворотну польську нотацію (RPN)
def до_rpn(вираз)
  вихід = []
  оператори = []

  вираз.scan(/\d+|\+|\-|\*|\/|/).each do |символ|
    if символ =~ /\d+/ # Якщо символ – це число
      вихід << символ
    elsif ['+', '-', '*', '/'].include?(символ) # Якщо символ – це оператор
      while оператори.any? && пріоритет(оператори.last) >= пріоритет(символ)
        вихід << оператори.pop
      end
      оператори << символ
    end
  end

  # Виводимо всі залишкові оператори
  вихід.concat(оператори.reverse)
  вихід.join(' ')
end

# Приклад
вираз = "2 + 1 * 4"
puts "Інфіксний вираз: #{вираз}"

```

```
puts "RPN вираз: #{до_rpn(вираз)}"
```

Вхідний код

```
9
10 # Функція для перетворення інфіксного виразу у зворотну польську нотацію (RPN)
11 def до_rpn(вираз)
12   вихід = []
13   оператори = []
14
15   вираз.scan(/\d+|\+|\-|\*|\/|/).each do |символ|
16     if символ =~ /\d+/ # Якщо символ - це число
17       вихід << символ
18     elsif ['+', '-', '*', '/'].include?(символ) # Якщо символ - це оператор
19       while оператори.any? && пріоритет(оператори.last) >= пріоритет(символ)
20         вихід << оператори.pop
21       end
22       оператори << символ
23     end
24   end
25
26   # Виводимо всі залишкові оператори
27   вихід.concat(оператори.reverse)
28   вихід.join(' ')
29 end
30
31 # Приклад
32 вираз = "2 + 1 * 4"
33 puts "Інфіксний вираз: #{вираз}"
34 puts "RPN вираз: #{до_rpn(вираз)}"
35
```

input

```
Інфіксний вираз: 2 + 1 * 4
RPN вираз: 2 1 4 * +

...Program finished with exit code 0
Press ENTER to exit console.
```

Малюнок 1 - результат