プログラミング言語 III(計算機言語 II) 2010 年度定期試験

実施日: 2011 年 1 月 26 日（水）
担当教員：国島丈生

I. Ruby プログラム

1. 以下の Ruby 式を評価した結果の値を示せ。(5 点×4)
   a) 1.to_s + 500.to_s
   b) /abe/ =~ “abdabcabeabf”
   c) [“50”, “100”, “1”, “200”].sort
   d) Array.new.push(1).push([1,2]).length
2. 自然数 n の b 進数表記（2 ≦ b ≦ 10）を文字列で返すメソッド itoa(n, b) を実装せよ。例えば itoa(100, 2) の結果は “1100100” となる。(20 点)
3. 文字列 s1, s2 に対して、s1 に含まれている文字をすべて s2 から除去した文字列を返すメソッド squeeze(s1, s2) を実装せよ。下記のコードの (*1) と書かれた部分のみ記述すれば良い。例えば squeeze(“abc”, “afbecdxyz”) の結果は “fedxyz” となる。(20 点)

```
def squeeze(s1, s2)
  array_s1 = s1.split(//)
  array_s2 = s2.split(//)
  result = ‘’
  # (*1)
  return result
end
```

II. Web アプリケーション

サーバ www.example.com 上で、次のプログラムを動かしている。このとき以下の問に答えよ。

```
#!/usr/bin/ruby

require 'webrick'

s = WEBrick::HTTPServer.new(
  :Port => 8000,
  :DocumentRoot => "/var/www/app"
)
```

```
s.mount_proc("/info") { |req, res|
  res.body = "This is WEBrick, made by Ruby."
}

s.mount_proc("/echo") { |req, res|
  # (*2)
}

Signal.trap("INT"){ s.shutdown }
s.start
```

1. ディレクトリ /var/www/app, /var/www/app/info には以下のファイルだけが置かれている。このとき、次の URI に HTTP GET メソッドを発行すると、どのようなステータスコードが返ってくるか答えよ。このサーバ上では、ほかの HTTP サーバプロセスは動作していないものとする。(5 点×4)

   /var/www/app:
   　　index.html　　logo.png
   /var/www/app/info:
   　　info.html

   a) http://www.example.com:8000/index.html
   b) http://www.example.com:8000/another.html
   c) http://www.example.com:8000/info
   d) http://www.example.com:8000/info/another.html
2. (*2) の部分に、以下のような処理をするコードを書き加えよ。(20 点)
   - リクエスト URI のパスが /echo で始まる場合は、そのパスをレスポンスボディとする（例: http://www.example.com:8000/echo/a/b/1 なら /echo/a/b/1 をレスポンスボディとする）
   - ただし、リクエスト URI のパスが /echo で始まり、かつ GET パラメータ length が 'true' なら、レスポンスボディにパスの長さも加える（例: http://www.example.com:8000/echo/a/b/1?length=true なら /echo/a/b/1 の長さが 11 なので、/echo/a/b/1, length: 11　をレスポンスボディとする）

解答例

I-1

a. “1500”
b. 6
c. ["1", "100", "200", "50"]
d. 2 （結果は [1, [1, 2]] という配列、すなわち 1 と [1, 2] からなる配列になります）

I-2

```
def itoa(n, b)
  result = ''
  while n != 0
    result = (n % b).to_s + result
    n /= b
  end
  result
end
```

I-3

解答 1（each を使ったもの）

```
def squeeze(s1, s2)
  array_s1 = s1.split(//)
  array_s2 = s2.split(//)
  result = ""
  array_s2.each do |c|
    if !array_s1.include? c
      result += c
    end
  end
  return result
end
```

解答 2（select, inject を使ったもの）

```
def squeeze2(s1, s2)
  array_s1 = s1.split(//)
```

```
  array_s2 = s2.split(//)
  result = ""
  result = array_s2.select { |c| !array_s1.include? c }.inject(result) { |result, c| result + c }
  return result
end
```

II-1

1. 200
2. 404
3. 200
4. 200（パスが /info で始まる URI はすべて mount_proc(“/info”) の部分のコードで処理されるため）

II-2

```
s.mount_proc("/echo") { |req, res|
  res.body = req.path
  if req.query['length'] == 'true'
    res.body += ", length: “ + req.path.length.to_s
  end
}
```