

令和3年度 基礎プログラミングII

出題者: 神田・広瀬・山本・西村・カロール・三浦

持ち込み: 教科書, 参考書, ノート, 電卓, 電子辞書専用機。携帯電話鳴動は即退室。

Type0 ← 解答の先頭にこの Type 番号を明記すること

第1問 以下の各問を答えよ。

- (1) 下の表の数値について基数変換し, 《ア》から《カ》までを埋めよ。ただし 0b, 0d, 0x は各々 2 進数, 10 進数, 16 進数を表す。

0b	0d	0x
《ア》	2022	《イ》
11110000	《ウ》	《エ》
《オ》	《カ》	beaf

- (2) 文字列 Jp21 の 4 文字について 16 進数の ASCII コードで表現せよ。なお, 0 は 0x30, A は 0x41, a は 0x61 である。

第2問 以下の数式を Ruby の記法にしたがって書き換えよ。ただし, Ruby では π は `Math::PI`, $\sqrt{x}$ は `Math.sqrt(x)` で表す。

- (1) $a \times b$ (2) πr^2
(3) $ax^2 + bx + c$ (4) $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$

第3問 変数 x に以下のようにハッシュを代入する。

```
x = {"りんご" => "apple", "みかん" => "orange", "なし" => "pear"}
```

このとき、以下の値は何になるか答えよ。

- (1) `x["みかん"]` (2) `x["バナナ"]`
(3) `x["なし"][3]` (4) `x.values[0]`
(5) `x.sort[0][1]`

第4問 プログラム f.rb で, $n > 0$ を満たす引数 n で, 以下のように再帰的に定義されるメソッド f を定義した。以下の問に答えよ。

```
def f(n)
  if n >= 1
    n + f(n-1)/2
  else
    0
  end
end
x = ARGV[0].to_i
printf("f(%d)=%f\n", x, f(x))
```

- (1) `f(1)` を求めよ。
(2) `f(2)` を求めよ。
(3) `f(3)` を求めるとき f メソッドは何回呼ばれるか。
(4) `./f.rb 1.5` と起動して `ARGV[0]` に "1.5" が渡るようにした。この時の `f(x)` の値はいくつか。

第5問 性別, 年齢, 身長 [cm], 体重 [kg] を格納した以下のような CSV ファイル dataset1.csv があり、全行で 4 項目が揃っているとする。

```
性別, 年齢, 身長, 体重
女, 22, 160.6, 67.2
男, 19, 171.6, 60.0
```

```
男, 21, 173.2, 70.8
女, 17, 149.4, 45.3
:
(以下同様のデータが続く)
```

以下のようにすると全ての身長数値、体重数値がそれぞれ配列の形で変数 h, w に入る。

```
ds1 = CSV.read("ds1.csv",
               :headers => true, :converters => :numeric)
h, w = ds1["身長"], ds1["体重"]
```

この状態から「`covar(h, w)`」のように呼び出し、身長と体重の共分散を求めるメソッドを作りたい。共分散は、身長、体重の i 番目の各データ h_i, w_i に対しそれぞれ $h_i - \bar{h}$ と $w_i - \bar{w}$ を求め、2 つを掛け合わせたものを全て合計し、最後にデータ数 n で割ったもの、すなわち

$$\frac{1}{n} \sum_{i=1}^n (h_i - \bar{h})(w_i - \bar{w}) \quad (\bar{h}, \bar{w} \text{ はそれぞれ身長、体重の平均値})$$

である。

```
def covar(x, y) # x, y はデータ数同じで全て数値の配列と仮定
  n = x.ア # 配列の要素数を n に代入
  xbar, ybar = x.sum/n, y.sum/n # 平均値を求める
  sum = イ
  0.upto(ウ) do |i| # 0 から 配列の末尾 まで繰り返す
    sum エ (x[i]-xbar)*(y[i]-ybar) # 計算結果を加算
  end
  オ
end
```

- (1) 空欄 《ア》から《オ》を埋めよ。
(2) リスト中 `x.sum`, `y.sum` とあるが、`sum` メソッドは何を求めるものと考えられるか。
(3) 受け取った 2 つの配列 `x`, `y` の要素数が違う可能性を考慮する必要が出た。個数が違う場合に `nil` を返すようにするには、(a) どの部分に、(b) どのような処理 を足せばよいか。
(4) 2 つの配列 `a1=[1,2,3]`, `a2=[4,5,6]` に対し、`a1.zip(a2)` と `zip` メソッドを用いると `[[1,4], [2,5], [3,6]]` のように 2 つの配列の各要素を順に取り出した配列を、さらに並べた配列の配列が得られる。また、`collect` メソッドは各要素に対して繰り返した値を集めた配列を返すため、`a1.collect{|e| 2*e}` とすると各要素全てを 2 倍した値を集めた配列 `[2,4,6]` を返す。これらを用いて `covar` を書き直したものが `cov2` である。空欄を埋めよ。

```
def cov2(x, y)
  n = x.length
  xbar, ybar = x.sum/n, y.sum/n
  x.zip(y).collect {|e|
    (e[0]-キ)*(キ-ybar)
  }.sum/n
end
```