# 本日の課題

文字列の配列が astring，比較方法の文字列 comp = "<"，および数 n が与えられたとき，astringの中で**出現頻度**が与えられた比較方法でその数と合う文字列を集めた配列を出力するプログラム kadai(astring, comp, n) を書け.

たとえば，kadai(astring, "<" 3) は，astringの文字列で出現頻度が3よりも小さい文字列によって構成された配列を値として返す．比較文字列としては，"<"，">"，"=="，"<="，">=" の5種類が使えるようにせよ.

テスト用の文字列の配列として，http://ideone.com/IXrRMT にある2つの文字列の配列を適宜使ってよい.

# ◇◇ kadai(astring, comp, n)の実行例 ◇◇

前述のテスト用文字列 samp1 を用いた場合，実行結果は以下のようになれば良い．

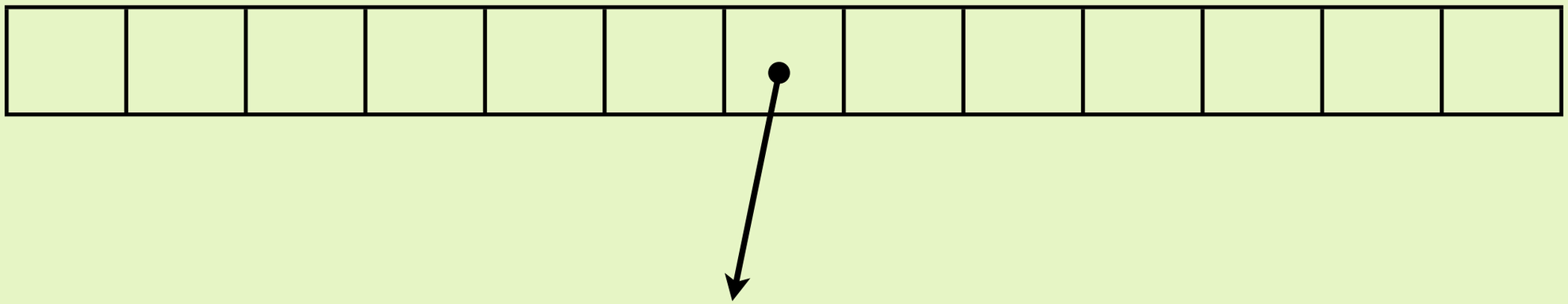
```
puts("[" + kadai(samp1, ">=", 5) + "] >= 5")
puts("[" + kadai(samp1, ">=", 8) + "] >= 8")
puts("[" + kadai(samp1, "==", 4) + "] == 4")
puts("[" + kadai(samp1, "<", 4) + "] < 2")
```

↓

```
[a,the,for,of,Greece,Mr,Tsipras,to,and] >= 5
[a,the,of,to,and] >= 8
[end,reforms,IMF,leaders,that,Athens,The] == 4
[reform,Greek,find,left,remain,Brussels,late,our,BBC,funds,week,Angela,bailout,earlier,put,had,towards,s,has,without,this,So,country,Chancellor,President,efforts,three,government,beginning,we,must,realised,total,talks,Earlier,resolve,when,move,fails,appears,prime,indicated,pensions,Minister,intensified,also,stuck,some,been,creditors,conceded,further,only,Damian,forward,Commission,bridge,In,writing,submitted,It,European,Prime,offer,ground,at,added,have,June,Thursday,German,correspondent,expected,repay,release,few,faces,said,be,believed,by,bundled,But,Eurozone,default,remaining,commits,We,primary,achieved,warning,viable,decided,with,relief,on,debts,differences,night,negotiations,economic,warned,sign,not,failure,Francois,work,progress,demand,but,Claude,Alexis,solution,told,Ms,would,sides,due,comments,adds,Jean,was,target,revised,agreed,Germany,want,plan,French,surplus,VAT,days,before,however,payment,minister,rejected,reports,there,off,making,all,return,chance,his,they,delaying,public,will,deal,Merkel,France,more,set,countenance,Grammaticas,after,until,he,growth,reporters,EU,meeting,debt,intensify,statement,After,reach,eurozone,Hollande,Last,paid,if,later,in,Juncker,constructive,is,it,little,hold,up] < 2
```

# 課題におけるハッシュテーブルの構成

ハッシュテーブルにそれぞれの単語を乗せて，さらに，出現回数を管理させる必要がある.



| Greece | 4 |
|---|---|
| France | 2 |
| Germany | 5 |

```
[["Greece", 4],
["France", 2],
["Germany", 5]]
```

# 解答例 (1)

まず，ハッシュ関数は授業で示したものと基本的に全く同じである．ここでは，大きさ1123のハッシュテーブルを作るので mmod = 1123とした．

```
function h(s){
    var mmod = 1123
    var m = 0
    for (var i = 0; i < s.length; i++){
        m = (m * 2341 + s.charCodeAt(i)) % mmod
    }
    return m
}
```

# 解答例 (2)

ハッシュテーブルを構成する．ハッシュテーブルに書き込むデータは文字列と頻度である．最初にテーブルに書き込むときは頻度１として，2回目以降は1ずつ増やす．

```
function construct(){
    var tbl = []
    for (var k = 0; k < data.length; k++){
        var s = data[k]
        var v = h(s)
        if (tbl[v] === undefined) {
            tbl[v] = [[s, 1]]
        } else {
            var ll = tbl[v]
            for (var el = 0; el < ll.length; el++){
                if (ll[el][0] == s){
                    ll[el][1] += 1
                    break
                }
            }
            if (el == ll.length) ll.push([s, 1])
        }
    }
    return tbl
}
```

# 解答例 (3)

与えられた条件に基づいて対象となる文字列を探し出す．基本的にハッシュテーブルを端から1つずつ調べて条件に合っていれば，resにその文字列を追加する．条件をチェックする関数はcheckである．

```
function scan(tbl, comp, n){
    function check(v, comp, n){
        if (comp == "<") return v < n
        else if (comp == ">") return v > n
        else if (comp == "==") return v == n
        else if (comp == ">=") return v >= n
        else if (comp == "<=") return v <= n
        else return null
    }
    var res = []
    for (var i = 0; i < tbl.length; i++){
        var ele = tbl[i]
        if (ele === undefined) continue;
        for (var j = 0; j < ele.length; j++){
            var [s, freq] = ele[j]
            if (check(freq, comp, n)){
                res.push(s)
            }
        }
    }
    return res
}
```

# 解答例 (4)

提出すべき関数kadaiは以下のとおり.

```
function kadai(data, comp, n){
    return scan(construct(), comp, n)
}
```

実際には前述の関数はこの関数kadaiの内部に定義することが望ましい. 実行例は以下のとおり：

```
puts("[" + kadai(samp1, ">=", 5).sort() + "] >= 5")
```

↓

```
[Greece,Mr,Tsipras,a,and,for,of,the,to] >= 5
```

```
puts("[" + kadai(samp1, ">=", 8).sort() + "] >= 8")
```

↓

```
[a,and,of,the,to] >= 8
```