



# 本日の課題



以下の関数 `rand_strings(n, m)` を実行すると、長さ `n` の小文字のアルファベットのランダムな文字列を `m` 個要素とするの配列が生成される。この配列には同じ文字列がいくつか含まれる可能性がある。重複を取り除いたとき、いくつ文字列が入っているかを計算しその数を返す関数 `kadai(strs)` をつくれ。

```
function rand_strings(n, m){
  var strs = [];
  function make_a_string(n){
    var pat = "abcdefghijklmnopqrstuvwxyz"
    var len = pat.length;
    var s = "";
    for (var j = 0; j < n; j++)
      s += pat.charAt(Math.floor(Math.random() * len));
    return s;
  }
  for (var i = 0; i < m; i++)
    strs.push(make_a_string(n));
  return strs
}
```

`kadai(rand_strings(5, 10000))`  
のようにしてテストせよ。

# 解答例

ハッシュ表にデータを順次いれていく。ただし、同じ文字列がすでに存在していれば、改めて入れることはしない。

```
function kadai(strs){
  var N = 20003;           ハッシュ関数
  var table = [];         は適当に作る
  function hash(s){
    var i, sum = 0;
    for (i = 0; i < s.length; i++){
      sum = (sum * 314 + s.charCodeAt(i)) % N;
    }
    return sum;
  }
  function put_elements_to_table(){
    var i, j;
    function put_an_element(h, s){
      var tableh = table[h];
      if (tableh == undefined) table[h] = [s];
      else{
        for (j = 0; j < tableh.length; j++){
          if (tableh[j] == s) return;
        }
        table[h].push(s);
      }
    }
    for (i = 0; i < strs.length; i++){
      put_an_element(hash(strs[i]), strs[i]);
    }
  }
}
```

```
var dat = rand_strings(5, 100000);
puts(dat.length);
puts(kadai(dat));
```



```
100000
99570
```

```
function count_elements(){
  var count = 0, i;
  for (i = 0; i < table.length; i++){
    if (table[i] != undefined)
      count += table[i].length;
  }
  return count;
}
put_elements_to_table();
return count_elements();
```