```
1:  #!/usr/bin/env python
2:  # -*- coding:utf-8 -*-
3:
4:  import re, sys
5:
6:  ##################正規表現　コンパイル
7:  #音節を抜き取る
8:  re_onsetsu = re.compile(u"[\u0F40-\u0FBC]+")
9:  #基字一文字
10: #(?!...)→もし ... が次に続くものとマッチしなければマッチします
11: re_onsetsu_kiji_only = re.compile(u"(?P<kiji>[\u0F40-\u0F6C])(?![\u0F71|\u0F90-\u0FBC])",re.X)
12:
13: ######################
14: #                       頭(ra)
15: re_onsetsu_atama_r = re.compile(u"(?P<atama>\u0F62)(?P<kiji>[\u0F90\u0F92\u0F94\u0F97\u0F99\u0F9F
    .  \u0FA1\u0FA3\u0FA6\u0FA8\u0FA9\u0FAB])", re.X)
16: #                       頭(la)
17: re_onsetsu_atama_l = re.compile(u"(?P<atama>\u0F63)(?P<kiji>[\u0F90\u0F92\u0F94\u0F95\u0F97\u0F9F
    .  \u0FA1\u0FA4\u0FA6\u0FB7])",re.X)
18: #                       頭(sa)
19: re_onsetsu_atama_s = re.compile(u"(?P<atama>\u0F66)(?P<kiji>[\u0F90\u0F92\u0F94\u0F99\u0F9F
    .  \u0FA1\u0FA3\u0FA4\u0FA6\u0FA8\u0FA9])",re.X)
20:
21: ######################
22: #                       #足(ya)
23: re_onsetsu_ashi_y = re.compile(u"(?P<kiji>[\u0F40\u0F41\u0F42\u0F54\u0F55\u0F56\u0F58])(?P<ashi>
    .  \u0FB1\u0FAD|\u0FB1)",re.X)
24: #                       #足(ra)
25: re_onsetsu_ashi_r = re.compile(u"(?P<kiji>[\u0F40\u0F41\u0F42\u0F4F
    .  \u0F50\u0F51\u0F54\u0F55\u0F56\u0F58\u0F64\u0F66\u0F67])(?P<ashi>\u0FB2\u0FAD|\u0FB2)",re.X)
26: #                       #足(la)
27: re_onsetsu_ashi_l = re.compile(u"(?P<kiji>[\u0F40\u0F42\u0F56\u0F5F\u0F62\u0F66])(?P<ashi>\u0FB3)",re.X)
28: #                       #足(wa)
29: re_onsetsu_ashi_w = re.compile(u"(?P<kiji>[\u0F40\u0F41\u0F42\u0F45\u0F49\u0F4F\u0F51\u0F59\u0F5A
    .  \u0F5E\u0F5F\u0F62\u0F63\u0F64\u0F66\u0F67])(?P<ashi>\u0FAD)?",re.X)
30:
31: ######################
32: #                       頭＋基＋足
33: re_onsetsu_atama_ashi = re.compile(u"""
34: (?P<atama>[\u0F62\u0F66])(?P<kiji>[\u0F90\u0F92\u0FA4\u0FA6\u0FA8])(?P<ashi>[\u0FB1\u0FB2])
35: """,re.X)
36:
37: ############                    #前置字(ga)の付く文字
38: re_onsetsu_zen_g_kiji_only = re.compile(u"(?P<zen>\u0F42)(?P<kiji>[\u0F45\u0F49\u0F4F
    .  \u0F51\u0F53\u0F59\u0F5E\u0F5F\u0F61\u0F64\u0F66])",re.X)
39:
40: ############                    前置字(da)の付く文字
41: #足の y　，　基(k,g,b,m
42: re_onsetsu_zen_d_ashi_y = re.compile(u"(?P<zen>\u0F51)(?P<kiji>[\u0F40\u0F42\u0F56\u0F58])(?P<ashi>
    .  \u0FB1\u0FAD|\u0FB1)",re.X)
43: #足の r　，基(k,p,b
44: re_onsetsu_zen_d_ashi_r = re.compile(u"(?P<zen>\u0F51)(?P<kiji>[\u0F40\u0F54\u0F56])(?P<ashi>
    .  \u0FB2\u0FAD|\u0FB2)",re.X)
45: ##
46: #基字のみ　，基(k,g,ng,p,b,m
```

```
47: re_onsetsu_zen_d_kiji_only = re.compile(u"(?P<zen>\u0F51)(?
    P<kiji>[\u0F40\u0F42\u0F44\u0F54\u0F56\u0F58])",re.X)
48:
49: ###########                    #前置字(ba)の付く文字
50: #頭と足
51: re_onsetsu_zen_b_atama_ashi = re.compile(u"(?P<zen_atama>\u0F56[\u0F62\u0F66])(?P<kiji>[\u0F90\u0F92])(?
    P<ashi>[\u0FB1\u0FB2])",re.X)
52:
53: #足の y　, 基(k,g
54: re_onsetsu_zen_b_ashi_y = re.compile(u"(?P<zen>\u0F56)(?P<kiji>[\u0F40\u0F42])(?P<ashi>\u0FB1\u0FAD|
    \u0FB1)",re.X)
55: #足の r　, 基(k,g
56: re_onsetsu_zen_b_ashi_r = re.compile(u"(?P<zen>\u0F56)(?P<kiji>[\u0F40\u0F42])(?P<ashi>\u0FB2\u0FAD|
    \u0FB2)",re.X)
57: #足の l　, 基(r,s
58: re_onsetsu_zen_b_ashi_l = re.compile(u"(?P<zen>\u0F56)(?P<kiji>[\u0F62\u0F66])(?P<ashi>\u0FB3)",re.X)
59:
60: #頭の r　, 基(k,g,ng,j,ny,t,d,n,ts,dz
61: re_onsetsu_zen_b_atama_r = re.compile(u"(?P<zen_atama>\u0F56\u0F62)(?
    P<kiji>[\u0F90\u0F92\u0F94\u0F97\u0F99\u0F9F\u0FA1\u0FA3\u0FA9\u0FAB])",re.X)
62: #頭の l　, 基(t
63: re_onsetsu_zen_b_atama_l = re.compile(u"(?P<zen_atama>\u0F56\u0F63)(?P<kiji>\u0F9F)",re.X)
64: #頭の s　, 基(k,g,ng,ny,t,d,n,ts
65: re_onsetsu_zen_b_atama_s = re.compile(u"(?P<zen_atama>\u0F56\u0F66)(?
    P<kiji>[\u0F90\u0F92\u0F94\u0F99\u0F9F\u0FA1\u0FA3\u0FA9])",re.X)
66:
67: #基字のみ(k,g,c,t,d,ts,zh,z,sh,s
68: re_onsetsu_zen_b_kiji_only = re.compile(u"(?P<zen>\u0F56)(?P<kiji>[\u0F40\u0F42\u0F45\u0F4F
    \u0F51\u0F59\u0F5E\u0F5F\u0F64\u0F66])",re.X)
69:
70: ###########                    #前置字(m)の付く文字
71: #足の y　, 基(kh,g
72: re_onsetsu_zen_m_ashi_y = re.compile(u"(?P<zen>\u0F58)(?P<kiji>[\u0F41\u0F42])(?P<ashi>\u0FB1\u0FAD|
    \u0FB1)",re.X)
73: #足の r　, 基(kh,g
74: re_onsetsu_zen_m_ashi_r = re.compile(u"(?P<zen>\u0F58)(?P<kiji>[\u0F41\u0F42])(?P<ashi>\u0FB2\u0FAD|
    \u0FB2)",re.X)
75:
76: #基字のみ(kh,g,ng,ch,j,ny,th,d,n,tsh,dz
77: re_onsetsu_zen_m_kiji_only = re.compile(u"(?P<zen>\u0F58)(?
    P<kiji>[\u0F41\u0F42\u0F44\u0F46\u0F47\u0F49\u0F50\u0F51\u0F53\u0F5A\u0F5B])",re.X)
78:
79: ###########                    #前置字('a)の付く文字
80: #足の y　, 基(kh,g,ph,b
81: re_onsetsu_zen_a_ashi_y = re.compile(u"(?P<zen>\u0F60)(?P<kiji>[\u0F41\u0F42\u0F55\u0F56])(?P<ashi>
    \u0FB1|\u0FB1\u0FAD)",re.X)
82: #足の r　, 基(kh,g,d,ph,b
83: re_onsetsu_zen_a_ashi_r = re.compile(u"(?P<zen>\u0F60)(?P<kiji>[\u0F41\u0F42\u0F51\u0F55\u0F56])(?P<ashi>
    \u0FB2\u0FAD|\u0FB2)",re.X)
84:
85: #       基字のみ(kh,g,ch,j,th,d,ph,b,tsh,dz
86: re_onsetsu_zen_a_kiji_only = re.compile(u"(?P<zen>\u0F60)(?
    P<kiji>[\u0F41\u0F42\u0F46\u0F47\u0F50\u0F51\u0F55\u0F56\u0F5A\u0F5B])",re.X)
87:
88: #       例外            #   子音字が続く場合、前の文字が基字である。
```

```
re_two_shion = re.compile(u"^(?P<kiji>[\u0F40-\u0F6C])(?P<kochi>[\u0F42\u0F44\u0F51\u0F53\u0F56\u0F58\u0F60\u0F62\u0F63\u0F66])$",re.X)

#　　例外　　#　　子音字３つで最後が「a」の場合、真ん中が基字　　←使っていない。
re_three_shion_last_a = re.compile(u"(?P<zen>[\u0F42\u0F51\u0F56\u0F58\u0F60])(?P<kiji>[\u0F40-\u0F6C])\u0F60",re.X)

#　　例外　　#　　子音字３つで最後が「s (\u0F66)」でない場合、真ん中が基字
re_three_shion_last_not_s = re.compile(u"^(?P<zen>[\u0F42\u0F51\u0F56\u0F58\u0F60])(?P<kiji>[\u0F40-\u0F6C])(?P<kochi>[\u0F42\u0F44\u0F51\u0F53\u0F56\u0F58\u0F60\u0F62\u0F63])$",re.X)

#　　例外　　#　　子音字３つで最後が「s」で、真ん中が「g,ng,b,m」でない場合、真ん中が基字
re_three_shion_last_s_center_not_gngbm = re.compile(u"^(?P<zen>[\u0F42\u0F51\u0F56\u0F58\u0F60])(?P<kiji>[\u0F40\u0F41\u0F45\u0F46\u0F47\u0F49\u0F4A\u0F4B\u0F4C\u0F4E\u0F4F
\u0F50\u0F51\u0F53\u0F54\u0F55\u0F59\u0F5A\u0F5B\u0F5D\u0F5E\u0F5F
\u0F60\u0F61\u0F62\u0F63\u0F64\u0F65\u0F66\u0F67\u0F68\u0F69\u0F6A\u0F6B\u0F6C])(?P<kochi>[\u0F66])$",re.X)

#　　例外　　#　　子音字３つで最後が「s」で、真ん中が「g,ng,b,m」である場合、最初の文字が基字
re_three_shion_last_m_center_gngbm = re.compile(u"^(?P<kiji>[\u0F40-\u0F6C])(?P<kochi>[\u0F42\u0F44\u0F56\u0F58])\u0F66$",re.X)

#　　例外　　#　個別の例外
##「dgs」は g が基字　&　「dms」は m が基字
re_reigai_kobetu_dgs_dms = re.compile(u"^(?P<zen>\u0F51)(?P<kiji>[\u0F42\u0F58])(?P<kochi>\u0F66)$",re.X)
##「ags」は g が基字
re_reigai_kobetu_ags = re.compile(u"^(?P<zen>\u0F60)(?P<kiji>\u0F42)(?P<kochi>\u0F66)$",re.X)
##「m,ng,s」は ng が基字
re_reigai_kobetu_mngs = re.compile(u"^(?P<zen>\u0F58)(?P<kiji>\u0F44)(?P<kochi>\u0F66)$",re.X)

#####残りを検索する
re_boin_kochiji = re.compile(u"""
(?P<boin>[\u0F72\u0F74\u0F7A\u0F7C\u0F7E])?　　#母音
(?P<kochi>[\u0F42\u0F44\u0F56\u0F58]\u0F66?|[\u0F51\u0F53\u0F60\u0F62\u0F63\u0F66]?[\u0F72\u0F74\u0F7A\u0F7C]?)　#後置+再後置+例外
""",re.X)

###############　サンスクリット語 (基字)(足+)(母音？)
re_skt = re.compile(u"""
(?P<kiji>[\u0F40-\u0F6C])(?P<ashi>[\u0F71|\u0F90-\u0FBC]+)(?P<boin>[\u0F71-\u0F83]?)
((?P<kiji2>[\u0F40-\u0F6C])(?P<ashi2>[\u0F71\u0F90-\u0FBC]?)(?P<boin2>[\u0F71-\u0F83]?))?
""",re.X)

#####################################################################
#マップオブジェクト
kiji_map = { u"\u0F40":u'\u0F40' , u"\u0F41":u'\u0F41' , u"\u0F42":u'\u0F42' , u"\u0F44":u'\u0F44' , u"\u0F45":u'\u0F45' , u"\u0F46":u'\u0F46' , u"\u0F47":u'\u0F47' , u"\u0F49":u'\u0F49' , u"\u0F4A":u'\u0F4A' , u"\u0F4B":u'\u0F4B' , u"\u0F4C":u'\u0F4C' , u"\u0F4E":u'\u0F4E' , u"\u0F4F":u'\u0F4F' , u"\u0F50":u'\u0F50' , u"\u0F51":u'\u0F51' , u"\u0F53":u'\u0F53' , u"\u0F54":u'\u0F54' , u"\u0F55":u'\u0F55' , u"\u0F56":u'\u0F56' , u"\u0F58":u'\u0F58' , u"\u0F59":u'\u0F59' , u"\u0F5A":u'\u0F5A' , u"\u0F5B":u'\u0F5B' , u"\u0F5D":u'\u0F5D' , u"\u0F5E":u'\u0F5E' , u"\u0F5F":u'\u0F5F' , u"\u0F60":u'\u0F60' , u"\u0F61":u'\u0F61' , u"\u0F62":u'\u0F62' , u"\u0F63":u'\u0F63' , u"\u0F64":u'\u0F64' , u"\u0F65":u'\u0F65' , u"\u0F66":u'\u0F66' , u"\u0F67":u'\u0F67' , u"\u0F68":u'\u0F68' , u"\u0F69":u'\u0F69' , u"\u0F6A":u'\u0F6A' , u"\u0F6B":u'\u0F6B' , u"\u0F6C":u'\u0F6C' , u"\u0F90":u'\u0F40' , u"\u0F92":u'\u0F42' , u"\u0F94":u'\u0F44' , u"\u0F95":u'\u0F45' , u"\u0F97":u'\u0F47' , u"\u0F99":u'\u0F49' , u"\u0F9F":u'\u0F4F' , u"\u0FA1":u'\u0F51' , u"\u0FA3":u'\u0F53' , u"\u0FA4":u'\u0F54' ,
```

```
u"\u0FA6":u'\u0F56' , u"\u0FA8":u'\u0F58' , u"\u0FA9":u'\u0F59' , u"\u0FAB":u'\u0F5B' , u"\u0FB7":u'\u0F67'
}
ashi_map = {
 u"\u0FAD":u'\u0F40' , u"\u0FB1":u'\u0F41' , u"\u0FB1\u0FAD":u'\u0F42' , u"\u0FB2":u'\u0F43' ,
u"\u0FB2\u0FAD":u'\u0F44' , u"\u0FB3":u'\u0F45' , u"\u0F71":"" , u"\u0F90":u"\u0F40" , u"\u0F91":u"\u0F41" ,
u"\u0F92":u'\u0F42'  , u"\u0F94":u'\u0F44'  , u"\u0F95":u'\u0F45'  , u"\u0F96":u'\u0F46'  ,
u"\u0F97":u'\u0F47'  , u"\u0F99":u'\u0F49'  , u"\u0F9F":u'\u0F4F'  , u"\u0FA0":u'\u0F50' , u"\u0FA1":u'\u0F51' ,
u"\u0FA3":u'\u0F53' , u"\u0FA4":u'\u0F54' , u"\u0FA5":u'\u0F55' , u"\u0FA6":u'\u0F56' , u"\u0FA8":u'\u0F58' ,
u"\u0FA9":u'\u0F59' , u"\u0FAA":u'\u0F5A' , u"\u0FAB":u'\u0F5B' , u"\u0FAE":u'\u0F5E' , u"\u0FAF":u'\u0F5F' ,
u"\u0FB0":u'\u0F60' , u"\u0FB4":u'\u0F64' , u"\u0FB6":u'\u0F66' , u"\u0FB7":u'\u0F67' , u"\u0FB8":u'\u0F68' ,
u"\u0FB9":u'\u0F69' , u"\u0FBA":u'\u0F6A' , u"\u0FBB":u'\u0F6B' , u"\u0FBC":u'\u0F6C'
 }
zen_map = { u"\u0F42":u'\u0F5F' , u"\u0F51":u'\u0F61' ,  u"\u0F56":u'\u0F62' ,  u"\u0F58":u'\u0F63' ,
u"\u0F60":u'\u0F64' }
atama_map = { u"\u0F62":u'\u0F65' , u"\u0F63":u'\u0F66' , u"\u0F66":u'\u0F67' }
zen_atama_map = { u"\u0F56\u0F62":u'\u0F68' , u"\u0F56\u0F63":u'\u0F69' , u"\u0F56\u0F66":u'\u0F6A' }
boin_map = {
 u"\u0F72" : u"\u0F21" , u"\u0F74" : u"\u0F22" , u"\u0F7A" : u"\u0F23" , u"\u0F7C" : u"\u0F24" , u"\u0F7E" :
u"\u0F25"
}
#後置字　再後置字　例外
kochi_map = {
 u"\u0F42" : u'\u0F40' , u"\u0F42\u0F66" : u'\u0F41' , u"\u0F44" : u'\u0F42' , u"\u0F44\u0F66" : u'\u0F43' ,
u"\u0F51" : u'\u0F44' , u"\u0F53" : u'\u0F45' , u'\u0F53\u0F72':u'\u0F46' , u'\u0F53\u0F74':u'\u0F47' ,
u'\u0F53\u0F7A':u'\u0F49' , u'\u0F53\u0F7C':u'\u0F4A' , u"\u0F56" : u'\u0F4B' , u"\u0F56\u0F66" : u'\u0F4C' ,
u"\u0F58" : u'\u0F4D' , u"\u0F58\u0F66" : u'\u0F4E' , u"\u0F60" : u'\u0F4F' , u"\u0F60\u0F72" : u'\u0F50' ,
u"\u0F60\u0F74" : u'\u0F51' , u"\u0F60\u0F7A" : u'\u0F52' , u"\u0F60\u0F7C" : u'\u0F53' , u"\u0F62" :
u'\u0F54' , u'\u0F62\u0F72':'\u0F55' , u'\u0F62\u0F74':'\u0F57' , u'\u0F62\u0F7A':'\u0F58' ,
u'\u0F62\u0F7C':'\u0F59' , u"\u0F63" : u'\u0F5A' , u"\u0F63\u0F72":u'\u0F5B' , u"\u0F63\u0F74":u'\u0F5C' ,
u"\u0F63\u0F7A":u'\u0F5D' , u"\u0F63\u0F7C":u'\u0F5E' , u"\u0F66" : u'\u0F5F' , u'\u0F66\u0F72':'\u0F60' ,
u'\u0F66\u0F74':'\u0F61' , u'\u0F66\u0F7A':'\u0F62' , u'\u0F66\u0F7C':'\u0F63' , u"\u0F7A" : u'\u0F64'  }

######################################################################
#書き出しのための関数
def kiji_pt(kiji,kiji_m):
        k_sk = ""
        m = kiji_m.match(kiji)
        if m:
                tango = m.groupdict()
                map = tango['kiji']
                k_sk = kiji_map[map]
                if tango.has_key('zen_atama') and tango['zen_atama']:
                        map = tango['zen_atama']
                        k_sk += zen_atama_map[map]
                if tango.has_key('atama') and tango['atama']:
                        map = tango['atama']
                        k_sk += atama_map[map]
                if tango.has_key('zen') and tango['zen']:
                        map = tango['zen']
                        k_sk += zen_map[map]
                if tango.has_key('ashi') and tango['ashi']:
                        map = tango['ashi']
                        k_sk += ashi_map[map]
        return k_sk

```

```
163:
164: def boin_kochiji(nokori):
165:         sk = ""
166:         sk2 = ""
167:         m = re_boin_kochiji.match(nokori)          #nokori　の中身を正規表現でmatchさせる
168:         if not m.group('boin'):                                #母音がなければ
169:                 sk = u'\u0F20'
170:         else:
171:                 sk = boin_map[m.group('boin')]             #re_boin_kochiji にHITした文字列リストの1番目
    を、boin_map に入れ、書き換え
172:         if m.group('kochi'):
173:                 sk2 = kochi_map[m.group('kochi')] #2番目を kochi に入れ、書き換え
174:         return sk + sk2
175:
176: def kiji_pt_reigai(rkiji, rkiji_m):
177:         rk_sk = ""
178:         m = rkiji_m.match(rkiji)
179:         if m:
180:                 tango = m.groupdict()
181:                 map = tango['kiji']
182:                 rk_sk = kiji_map[map]
183:                 if tango.has_key('zen'):
184:                         map = tango['zen']
185:                         rk_sk += zen_map[map]
186:                 if not tango.has_key('boin'):
187:                         rk_sk += u'\u0F20'
188:                 if tango.has_key('kochi'):
189:                         map = tango['kochi']
190:                         rk_sk += kochi_map[map]
191:         return rk_sk
192:
193: ################################################
194: #正規表現の「.match」
195: #どの条件にHITしたかを書く
196: def one_onsetsu(m):
197:         one_onsetsu = m.group()             #ＨＩＴした文字列のリストを変数に代入
198: #       ※例外#####           子音字が二つ続く場合、前が基字       ※例外
199:         mx = re_two_shion.match(one_onsetsu)
200:         if mx:
201:                 rkiji_m = re_two_shion
202:                 rkiji = one_onsetsu[mx.start():]
203:                 sw1 = kiji_pt_reigai(rkiji, rkiji_m)
204:                 return sw1
205: ############                          前ｂ＋頭＋基＋足
206:         mx = re_onsetsu_zen_b_atama_ashi.match(one_onsetsu)                #この正規表現にmatchしたらば、↓
207:         if mx:
208:                 kiji_m = re_onsetsu_zen_b_atama_ashi
209:                 kiji = one_onsetsu[mx.start():]
210:                 sw1 = kiji_pt(kiji,kiji_m)
211:                 nokori = one_onsetsu[mx.end():]             #match したものの後ろを nokori に入れる
212:                 sw2 = boin_kochiji(nokori)                  #boin_kochijiの関数を実行し、sw2に入れる
213:                 return sw1 + sw2
214: ########################################
215: ######                          個別の例外                              #######
216:         mx = re_reigai_kobetu_dgs_dms.match(one_onsetsu)
```

```
217:         if mx:
218:                 rkiji_m = re_reigai_kobetu_dgs_dms
219:                 rkiji = one_onsetsu[mx.start():]
220:                 sw1 = kiji_pt_reigai(rkiji, rkiji_m)
221:                 return sw1
222:         mx = re_reigai_kobetu_ags.match(one_onsetsu)
223:         if mx:
224:                 rkiji_m = re_reigai_kobetu_ags
225:                 rkiji = one_onsetsu[mx.start():]
226:                 sw1 = kiji_pt_reigai(rkiji, rkiji_m)
227:                 return sw1
228:         mx = re_reigai_kobetu_mngs.match(one_onsetsu)
229:         if mx:
230:                 rkiji_m = re_reigai_kobetu_mngs
231:                 rkiji = one_onsetsu[mx.start():]
232:                 sw1 = kiji_pt_reigai(rkiji, rkiji_m)
233:                 return sw1
234: ########################################
235: #       ※例外###### 子音字３つで最後が「s」でない場合、真ん中が基字　※例外
236:         mx = re_three_shion_last_not_s.match(one_onsetsu)
237:         if mx:
238:                 rkiji_m = re_three_shion_last_not_s
239:                 rkiji = one_onsetsu[mx.start():]
240:                 sw1 = kiji_pt_reigai(rkiji, rkiji_m)
241:                 return sw1
242: #       ※例外###### 子音字３つで最後が「s」で、真ん中が「g,ng,b,m」でない場合、真ん中が基字　※例外
243:         mx = re_three_shion_last_s_center_not_gngbm.match(one_onsetsu)
244:         if mx:
245:                 rkiji_m = re_three_shion_last_s_center_not_gngbm
246:                 rkiji = one_onsetsu[mx.start():]
247:                 sw1 = kiji_pt_reigai(rkiji, rkiji_m)
248:                 return sw1
249: #       ※例外###### 子音字３つで最後が「s」で、真ん中が「g,ng,b,m」である場合、最初の文字が基字　※例外
250:         mx = re_three_shion_last_m_center_gngbm.match(one_onsetsu)
251:         if mx:
252:                 rkiji_m = re_three_shion_last_m_center_gngbm
253:                 rkiji = one_onsetsu[mx.start():]
254:                 sw1 = kiji_pt_reigai(rkiji, rkiji_m)
255:                 return sw1
256: ############                            前ｂ＋頭＋基
257:         mx = re_onsetsu_zen_b_atama_s.match(one_onsetsu)
258:         if mx:
259:                 kiji_m = re_onsetsu_zen_b_atama_s
260:                 kiji = one_onsetsu[mx.start():]
261:                 sw1 = kiji_pt(kiji,kiji_m)
262:                 nokori = one_onsetsu[mx.end():]
263:                 sw2 = boin_kochiji(nokori)
264:                 return sw1 + sw2
265:         mx = re_onsetsu_zen_b_atama_l.match(one_onsetsu)
266:         if mx:
267:                 kiji_m = re_onsetsu_zen_b_atama_l
268:                 kiji = one_onsetsu[mx.start():]
269:                 sw1 = kiji_pt(kiji,kiji_m)
270:                 nokori = one_onsetsu[mx.end():]
271:                 sw2 = boin_kochiji(nokori)
```

```
272:                return sw1 + sw2
273:        mx = re_onsetsu_zen_b_atama_r.match(one_onsetsu)
274:        if mx:
275:                kiji_m = re_onsetsu_zen_b_atama_r
276:                kiji = one_onsetsu[mx.start():]
277:                sw1 = kiji_pt(kiji,kiji_m)
278:                nokori = one_onsetsu[mx.end():]
279:                sw2 = boin_kochiji(nokori)
280:                return sw1 + sw2
281: ############                                  前a＋基＋足
282:        mx = re_onsetsu_zen_a_ashi_r.match(one_onsetsu)
283:        if mx:
284:                kiji_m = re_onsetsu_zen_a_ashi_r
285:                kiji = one_onsetsu[mx.start():]
286:                sw1 = kiji_pt(kiji,kiji_m)
287:                nokori = one_onsetsu[mx.end():]
288:                sw2 = boin_kochiji(nokori)
289:                return sw1 + sw2
290:        mx = re_onsetsu_zen_a_ashi_y.match(one_onsetsu)
291:        if mx:
292:                kiji_m = re_onsetsu_zen_a_ashi_y
293:                kiji = one_onsetsu[mx.start():]
294:                sw1 = kiji_pt(kiji,kiji_m)
295:                nokori = one_onsetsu[mx.end():]
296:                sw2 = boin_kochiji(nokori)
297:                return sw1 + sw2
298: ############                                  前m＋基＋足
299:        mx = re_onsetsu_zen_m_ashi_r.match(one_onsetsu)
300:        if mx:
301:                kiji_m = re_onsetsu_zen_m_ashi_r
302:                kiji = one_onsetsu[mx.start():]
303:                sw1 = kiji_pt(kiji,kiji_m)
304:                nokori = one_onsetsu[mx.end():]
305:                sw2 = boin_kochiji(nokori)
306:                return sw1 + sw2
307:        mx = re_onsetsu_zen_m_ashi_y.match(one_onsetsu)
308:        if mx:
309:                kiji_m = re_onsetsu_zen_m_ashi_y
310:                kiji = one_onsetsu[mx.start():]
311:                sw1 = kiji_pt(kiji,kiji_m)
312:                nokori = one_onsetsu[mx.end():]
313:                sw2 = boin_kochiji(nokori)
314:                return sw1 + sw2
315: ############                                  前置字b　足の～
316:        mx = re_onsetsu_zen_b_ashi_l.match(one_onsetsu)
317:        if mx:
318:                kiji_m = re_onsetsu_zen_b_ashi_l
319:                kiji = one_onsetsu[mx.start():]
320:                sw1 = kiji_pt(kiji,kiji_m)
321:                nokori = one_onsetsu[mx.end():]
322:                sw2 = boin_kochiji(nokori)
323:                return sw1 + sw2
324:        mx = re_onsetsu_zen_b_ashi_r.match(one_onsetsu)
325:        if mx:
326:                kiji_m = re_onsetsu_zen_b_ashi_r
```

```
327:                 kiji = one_onsetsu[mx.start():]
328:                 sw1 = kiji_pt(kiji,kiji_m)
329:                 nokori = one_onsetsu[mx.end():]
330:                 sw2 = boin_kochiji(nokori)
331:                 return sw1 + sw2
332:         mx = re_onsetsu_zen_b_ashi_y.match(one_onsetsu)
333:         if mx:
334:                 kiji_m = re_onsetsu_zen_b_ashi_y
335:                 kiji = one_onsetsu[mx.start():]
336:                 sw1 = kiji_pt(kiji,kiji_m)
337:                 nokori = one_onsetsu[mx.end():]
338:                 sw2 = boin_kochiji(nokori)
339:                 return sw1 + sw2
340: ##########                                  前置字d　足
341:         mx = re_onsetsu_zen_d_ashi_r.match(one_onsetsu)
342:         if mx:
343:                 kiji_m = re_onsetsu_zen_d_ashi_r
344:                 kiji = one_onsetsu[mx.start():]
345:                 sw1 = kiji_pt(kiji,kiji_m)
346:                 nokori = one_onsetsu[mx.end():]
347:                 sw2 = boin_kochiji(nokori)
348:                 return sw1 + sw2
349:         mx = re_onsetsu_zen_d_ashi_y.match(one_onsetsu)
350:         if mx:
351:                 kiji_m = re_onsetsu_zen_d_ashi_y
352:                 kiji = one_onsetsu[mx.start():]
353:                 sw1 = kiji_pt(kiji,kiji_m)
354:                 nokori = one_onsetsu[mx.end():]
355:                 sw2 = boin_kochiji(nokori)
356:                 return sw1 + sw2
357: ##########                                  頭＋基＋足
358:         mx = re_onsetsu_atama_ashi.match(one_onsetsu)
359:         if mx:
360:                 kiji_m = re_onsetsu_atama_ashi
361:                 kiji = one_onsetsu[mx.start():]
362:                 sw1 = kiji_pt(kiji,kiji_m)
363:                 nokori = one_onsetsu[mx.end():]
364:                 sw2 = boin_kochiji(nokori)
365:                 return sw1 + sw2
366:
367: ##########                                  前置字a　基字のみ
368:         mx = re_onsetsu_zen_a_kiji_only.match(one_onsetsu)
369:         if mx:
370:                 kiji_m = re_onsetsu_zen_a_kiji_only
371:                 kiji = one_onsetsu[mx.start():]
372:                 sw1 = kiji_pt(kiji,kiji_m)
373:                 nokori = one_onsetsu[mx.end():]
374:                 sw2 = boin_kochiji(nokori)
375:                 return sw1 + sw2
376: ##########                                  前置字m　基字のみ
377:         mx = re_onsetsu_zen_m_kiji_only.match(one_onsetsu)
378:         if mx:
379:                 kiji_m = re_onsetsu_zen_m_kiji_only
380:                 kiji = one_onsetsu[mx.start():]
381:                 sw1 = kiji_pt(kiji,kiji_m)
```

```
382:                 nokori = one_onsetsu[mx.end():]
383:                 sw2 = boin_kochiji(nokori)
384:                 return sw1 + sw2
385: ##########                                  前置字b　基字のみ
386:         mx = re_onsetsu_zen_b_kiji_only.match(one_onsetsu)
387:         if mx:
388:                 kiji_m = re_onsetsu_zen_b_kiji_only
389:                 kiji = one_onsetsu[mx.start():]
390:                 sw1 = kiji_pt(kiji,kiji_m)
391:                 nokori = one_onsetsu[mx.end():]
392:                 sw2 = boin_kochiji(nokori)
393:                 return sw1 + sw2
394: ##########                                  前置字d　基字のみ
395:         mx = re_onsetsu_zen_d_kiji_only.match(one_onsetsu)
396:         if mx:
397:                 kiji_m = re_onsetsu_zen_d_kiji_only
398:                 kiji = one_onsetsu[mx.start():]
399:                 sw1 = kiji_pt(kiji,kiji_m)
400:                 nokori = one_onsetsu[mx.end():]
401:                 sw2 = boin_kochiji(nokori)
402:                 return sw1 + sw2
403: ##########                                  前置字g　基字のみ
404:         mx = re_onsetsu_zen_g_kiji_only.match(one_onsetsu)
405:         if mx:
406:                 kiji_m = re_onsetsu_zen_g_kiji_only
407:                 kiji = one_onsetsu[mx.start():]
408:                 sw1 = kiji_pt(kiji,kiji_m)
409:                 nokori = one_onsetsu[mx.end():]
410:                 sw2 = boin_kochiji(nokori)
411:                 return sw1 + sw2
412: ###########                           頭＋基
413:         mx = re_onsetsu_atama_s.match(one_onsetsu)
414:         if mx:
415:                 kiji_m = re_onsetsu_atama_s
416:                 kiji = one_onsetsu[mx.start():]
417:                 sw1 = kiji_pt(kiji,kiji_m)
418:                 nokori = one_onsetsu[mx.end():]
419:                 sw2 = boin_kochiji(nokori)
420:                 return sw1 + sw2
421:         mx = re_onsetsu_atama_l.match(one_onsetsu)
422:         if mx:
423:                 kiji_m = re_onsetsu_atama_l
424:                 kiji = one_onsetsu[mx.start():]
425:                 sw1 = kiji_pt(kiji,kiji_m)
426:                 nokori = one_onsetsu[mx.end():]
427:                 sw2 = boin_kochiji(nokori)
428:                 return sw1 + sw2
429:         mx = re_onsetsu_atama_r.match(one_onsetsu)
430:         if mx:
431:                 kiji_m = re_onsetsu_atama_r
432:                 kiji = one_onsetsu[mx.start():]
433:                 sw1 = kiji_pt(kiji,kiji_m)
434:                 nokori = one_onsetsu[mx.end():]
435:                 sw2 = boin_kochiji(nokori)
436:                 return sw1 + sw2
```

```
437: ###########                          基＋足
438:         mx = re_onsetsu_ashi_l.match(one_onsetsu)
439:         if mx:
440:                 kiji_m = re_onsetsu_ashi_l
441:                 kiji = one_onsetsu[mx.start():]
442:                 sw1 = kiji_pt(kiji,kiji_m)
443:                 nokori = one_onsetsu[mx.end():]
444:                 sw2 = boin_kochiji(nokori)
445:                 return sw1 + sw2
446:         mx = re_onsetsu_ashi_r.match(one_onsetsu)
447:         if mx:
448:                 kiji_m = re_onsetsu_ashi_r
449:                 kiji = one_onsetsu[mx.start():]
450:                 sw1 = kiji_pt(kiji,kiji_m)
451:                 nokori = one_onsetsu[mx.end():]
452:                 sw2 = boin_kochiji(nokori)
453:                 return sw1 + sw2
454:         mx = re_onsetsu_ashi_y.match(one_onsetsu)
455:         if mx:
456:                 kiji_m = re_onsetsu_ashi_y
457:                 kiji = one_onsetsu[mx.start():]
458:                 sw1 = kiji_pt(kiji,kiji_m)
459:                 nokori = one_onsetsu[mx.end():]
460:                 sw2 = boin_kochiji(nokori)
461:                 return sw1 + sw2
462:         mx = re_onsetsu_ashi_w.match(one_onsetsu)
463:         if mx:
464:                 kiji_m = re_onsetsu_ashi_w
465:                 kiji = one_onsetsu[mx.start():]
466:                 sw1 = kiji_pt(kiji,kiji_m)
467:                 nokori = one_onsetsu[mx.end():]
468:                 sw2 = boin_kochiji(nokori)
469:                 return sw1 + sw2
470: ############################         基字
471:         mx = re_onsetsu_kiji_only.match(one_onsetsu)
472:         if mx:
473:                 kiji_m = re_onsetsu_kiji_only
474:                 kiji = one_onsetsu[mx.start():]
475:                 sw1 = kiji_pt(kiji,kiji_m)
476:                 nokori = one_onsetsu[mx.end():]
477:                 sw2 = boin_kochiji(nokori)
478:                 return sw1 + sw2
479:
480: ############################         サンスクリット語
481:         mx = re_skt.match(one_onsetsu)
482:         if mx:
483:                 skt = ""
484:                 tango = mx.groupdict()
485:                 map = tango['kiji']
486:                 skt = kiji_map[map]
487:                 map = tango['ashi']
488:                 for ch in map:
489:                         skt += ashi_map[ch]
490:                 if tango['boin']:
491:                         map = tango['boin']
```

```
492:                    skt += boin_map[map]
493:            else:
494:                    skt += u'\u0F20'
495:            if tango['kiji2']:
496:                    map = tango['kiji2']
497:                    skt += kiji_map[map]
498:            if tango['ashi2']:
499:                    map = tango['ashi2']
500:                    for ch in map:
501:                            skt += ashi_map[ch]
502:            if tango['boin2']:
503:                    map = tango['boin2']
504:                    skt += boin_map[map]
505:            return skt
506:
507:
508: ######################################################
509: #part3          ファイル・行の読み込み
510:
511: #subにマッチした文字列を関数「one_onsetsu」に入れる
512: #「for 変数 in オブジェクト
513: #                実行される処理・・・」
514: #       →オブジェクトから要素を一つずつ変数に代入しながら繰り返しを行う
515: if len(sys.argv) == 1:
516:         fin = sys.stdin
517: else:
518:         fin = open(sys.argv[1], "rU")
519: debag = False
520: i = 0
521: for line in fin:
522:         line = unicode(line, "utf-8") #読み込んだ行をUnicodeに変換
523:         line2 = re_onsetsu.sub(one_onsetsu, line)
524:         if debag:
525:                 print line2.encode('utf-8') + "," + line.encode("utf-8")
526:         else:
527:                 print line2.encode('utf-8'),
```