```
#!/usr/bin/env python
#                                -*- Mode: Python -*-

R"""bbdbpa converts a bbdb file to a pilot-addresses file

Usage: bbdbpa [[bbdb-file] pa-file]

bbdbpa converts a bbdb file to a pilot-addresses file (pa-file). The pa-file can be used to install the addresses in your palm OS device with the pilot-addresses program (part of the pilot-link package [1]).

(C) Copyright Anders Andersen 2000 (see COPYING for details).

[1] ftp://ryeham.ee.ryerson.ca/pub/PalmOS/
"""

# Inserted by rcs (cvs)
rcs_author = "$Author: aa $"
rcs_date = "$Date: 2000/02/17 10:22:56 $"
rcs_revision = "$Revision: 1.5 $"

# Load some module
import sys, os, re, string

# Arguments (input and output)
input = ofnn = None
if len(sys.argv) > 3:
    print "bbdbpa: max 2 argumnets.\n"
    print __doc__,
    sys.exit(1)
elif len(sys.argv) == 3:
    ifn = sys.argv[1]
    try:
        input = open(ifn)
    except IOError:
        print "bbdbpa: unable to open %s.\n" % (ifn,)
        print __doc__,
        sys.exit(1)
    ofnn = 2
elif len(sys.argv) == 2:
    ofnn = 1

# Open (create) output file
if ofnn:
    try:
        output = open(sys.argv[ofnn], 'w')
    except IOError:
        print "bbdbpa: unable to create %s.\n" % (sys.argv[ofnn],)
        print __doc__,
        sys.exit(1)
else:
    output = sys.stdout

# Open default input (if no file was given)
if not input:
    try:
        fn = "%s/.bbdb" % (os.environ['HOME'],)
    except KeyError:
        fn = ".bbdb"
    try:
        input = open(fn)
```

```
    except IOError:
        input = sys.stdin

# The fields in the BBDB database
sf = 0        # Sort field (generated from ln, fn and com)
fn = 1        # First name ("fn")
ln = 2        # Last name ("ln")
aka = 3       # Aliases (list of "aka")
com = 4       # Company ("com")
ph = 5        # Phone numbers (list of ["type" "num"])
addr = 6      # Addresses (list of ["type" "adr"])
net = 7       # Email addresses (list of "net")
note = 8      # Notes/timestamp (user specified fields, list of lists?)
opt = 9       # ?

# A regexp for each bbdb line
bbdbline = re.compile(
    r'^\[(nil|".*")' +           # fn
    r'\s+(nil|".*")' +           # ln
    r'\s+(nil|\(".*"\))' +       # aka
    r'\s+(nil|".*")' +           # com
    r'\s+(nil|\(\["*"\]\))' +    # ph
    r'\s+(nil|\(\[".*\]\))' +    # addr
    r'\s+(nil|\(".*"\))' +       # net
    r'\s+(nil|\(\(.*\)\))' +     # note/timestamp
    r'\s+(nil)\]$')               # ?

charmap = {
    "ð" : "D",
    "é" : "E",
    "É" : "E",
    "æ" : "ZZ",
    "Æ" : "ZZ",
    "ø" : "ZZZ",
    "Ø" : "ZZZ",
    "ö" : "ZZZ",
    "Ö" : "ZZZ",
    "å" : "ZZZZ",
    "Å" : "ZZZZ",
    "AA": "ZZZZ"}

statelist = ["CA"]       # Not complete yet

zipexp = re.compile(r'(.*)( (\(\d+ \d+\)|\d+))$')
zipnum = re.compile(r'\((\d+) (\d+)\)')

def mapchars(str):
    for c in charmap.keys():
        if string.find(str, c) != -1:
            str = string.replace(str, c, charmap[c])
    return str

def mkcmpstr(addr):
    return mapchars(
        string.upper(addr[ln][1:-1] + addr [fn][1:-1] + addr[com][1:-1]))

def cmpaddr(addr1, addr2):
    cmpstr1 = mkcmpstr(addr1)
```

```
    cmpstr2 = mkcmpstr(addr2)
    if cmpstr1 < cmpstr2:
        return -1
    elif cmpstr1 > cmpstr2:
        return 1
    else:
        return 0

def unzipaddr(str):
    zip = ""
    if zipexp.match(str):
        m = zipexp.match(str)
        str = m.group(1)
        if zipnum.match(m.group(3)):
            zip = "%s-%s" % zipnum.match(m.group(3)).groups()
        elif m.group(3) != "0":
            zip = m.group(3)
    return zip, str

def splitstrlist(str):
    if str == "nil":
        return []
    else:
        return string.split(str[2:-2], '" "')

def splitphlist(str):
    if str == "nil":
        return []
    else:
        phlist = string.split(str[2:-2], '] [')
        for i in range(len(phlist)):
            zip, phstr = unzipaddr(phlist[i])
            phlist[i] = splitstrlist("(" + phstr + ")")
            phlist[i].append(zip)
        return phlist

# Read the bbdb file
addrlist = []
lines = input.readlines()
for line in lines:

    # Ignore none-records (comments, ...)
    if line[0] != '[':
        continue

    # Parse (split) a bbdb line
    bbdbmatch = bbdbline.match(line)
    if not bbdbmatch:
        print "bbdbpa: unable to parse bbdbline:\n%s\n" % (line,)
        print __doc__,
        sys.exit(1)

    # Initialize list
    bbdblist = ["]
    for i in range(1,len(bbdbmatch.groups())):
        bbdblist.append(bbdbmatch.group(i))

    # Clean up fn, ln, com
```

```
    for i in [fn, ln, com]:                                                    179
        if bbdblist[i] == "nil":                                               180
            bbdblist[i] = '""'                                                 181
                                                                               182
    # Convert phone numbers (max 4 used)                                       183
    bbdblist[ph] = splitphlist(bbdblist[ph])                                   184
    for i in range(len(bbdblist[ph])):                                         185
        tmpphn = string.replace(bbdblist[ph][i][1], " ", "")                   186
        tmppht = string.lower(bbdblist[ph][i][0])                              187
        bbdblist[ph][i] = '"%s (%s)"' % (tmpphn, tmppht)                       188
    if len(bbdblist[ph]) < 4:                                                  189
        for i in range(len(bbdblist[ph]), 4):                                  190
            bbdblist[ph].append('""')                                          191
                                                                               192
    # Convert addresses                                                        193
    addrs = splitphlist(bbdblist[addr])                                        194
    if len(addrs) > 0:                                                         195
        bbdblist[addr] = [[''] * 5] * len(addrs)                               196
    else:                                                                      197
        bbdblist[addr] = [['""'] * 5]                                          198
    for i in range(len(addrs)):                                                199
                                                                               200
        # (Street) Address part                                                201
        bbdblist[addr][i][0] = ''                                              202
        for j in [1, 2, 3]:                                                    203
            if addrs[i][j]:                                                    204
                bbdblist[addr][i][0] = bbdblist[addr][i][0] + addrs[i][j] +"\n"205
        bbdblist[addr][i][0] = '"' + bbdblist[addr][i][0] + '"'                206
        if bbdblist[addr][i][0][-2] == "\n":                                   207
            bbdblist[addr][i][0] = bbdblist[addr][i][0][:-2] + '"'             208
        bbdblist[addr][i][1] = '"' + addrs[i][4] + '"'                         209
                                                                               210
        # State and country                                                    211
        if addrs[i][5] in statelist:                                           212
            bbdblist[addr][i][2] = '"' + addrs[i][5] + '"'                     213
            bbdblist[addr][i][4] = '""'                                        214
        else:                                                                  215
            bbdblist[addr][i][4] = '"' + addrs[i][5] + '"'                     216
            bbdblist[addr][i][2] = '""'                                        217
                                                                               218
        # Zip code                                                             219
        bbdblist[addr][i][3] = '"' + addrs[i][6] + '"'                         220
                                                                               221
    # Convert email addresses                                                  222
    bbdblist[net] = '"%s"' % (                                                 223
        string.join(splitstrlist(bbdblist[net]), ", "),)                       224
                                                                               225
    # Append this address to address list                                      226
    addrlist.append(bbdblist)                                                  227
                                                                               228
# Sort address list                                                            229
addrlist.sort(cmpaddr)                                                         230
                                                                               231
# Print pa file                                                                232
for addritem in addrlist:                                                      233
    output.write(                                                              234
        "%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s\n" % (    235
            addritem[ln],                                                      236
```

```
addritem[fn],
'""',                          # title
addritem[com],
addritem[ph][0],
addritem[ph][1],
addritem[ph][2],
addritem[ph][3],
addritem[net],
addritem[addr][0][0],
addritem[addr][0][1],
addritem[addr][0][2],
addritem[addr][0][3],
addritem[addr][0][4],
'""',                          # custom 1
'""',                          # custom 2
'""',                          # custom 3
'""',                          # custom 4
'""',                          # note
'"0"'))
```