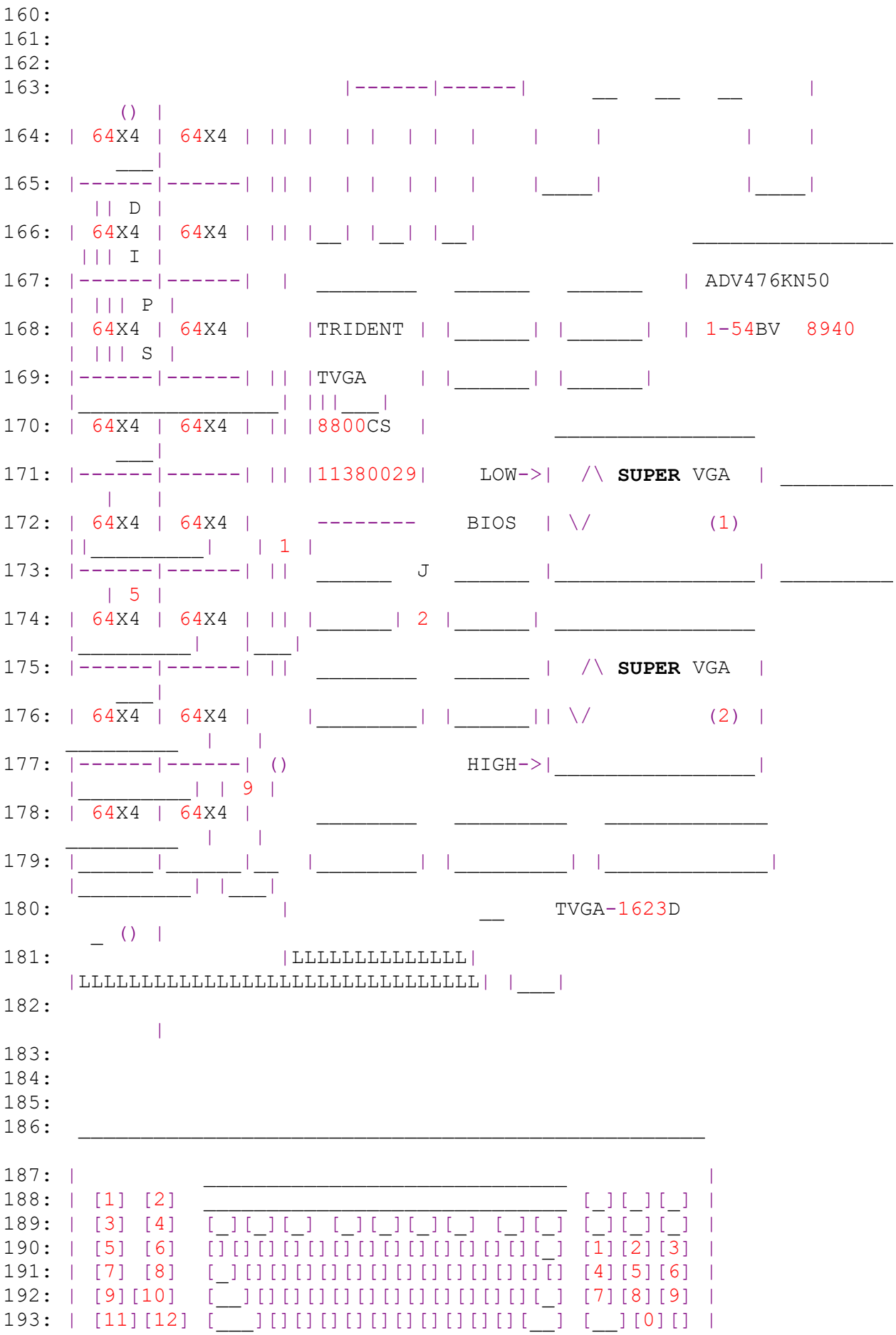
```
 1: ************************************************
 2:   maXbox Starter 97
 3: ************************************************
 4:
 5:  Operating System Services OSS
 6: ------------------------------------
 7:  Max Kleiner
 8:
 9:  //2 Worte werden Dir im Leben Türen öffnen - "ziehen" und
    "stossen".
10:  Two words will open many doors for you in life - "pull" and
    "push".
11:
12:  This report will show a few important routines an operating
    system needs. An Operating System (OS) is an interface between a
    user and hardware. An OS is a software and an API which performs
    all the basic tasks like file management, memory management,
    process management, handling input and output including graphics,
    and controlling peripheral devices such as disk drives and
    printers.
13:
14:                  ____  ______.
15:   _____ _____   \    \/   /\_ |__    _______  ___
16:  /     \\__  \   \     /    | __ \  /  _ \  \/  /
17: |  Y Y  \/ __ \_/     \    | \_\ (  <_> >    <
18: |__|_|  (____  /___/\  \   |___  /\____/__/\_ \
19:       \/     \/      \_/       \/            \/
20:
21: https://www.tutorialspoint.com/operating_system/os_quick_guide.htm
22: https://github.com/maxkleiner/argos-translate
23: https://github.com/LibreTranslate/LibreTranslate#mirrors
24:
25: The best way to organize OS-routines is object oriented. A lot of
    OS-APIs are in fact object-oriented, for example everything that
    returns a handle. OOP has nothing to do with not being able to
    unsafe operations in your programming language. I think what you
    are really asking is: Can you write an OS in a language with
    automatic memory management and memory-safety?
26:
27: An example for returning a handle is the function
    GetForegroundWindow():
28:
29: function GetForegroundWindow: THandle;
30:
31: And another is the counterpart to set the foreground but you need
    the handle:
32:
33: function SetForegroundWindow (hwnd : THandle) : BOOL
34:
35: Makes the specified window the current foreground window and gives
    it the focus. This function should only be used with windows which
    your program owns. Of course this function should be used with
    caution, since the user usually doesn
    window to change unexpectedly. The function tells Windows to
    somehow draw the user's attention to the window, such as by
    flashing its icon in the taskbar. The function returns 1 if
    successful, or 0 if an error occured.
```

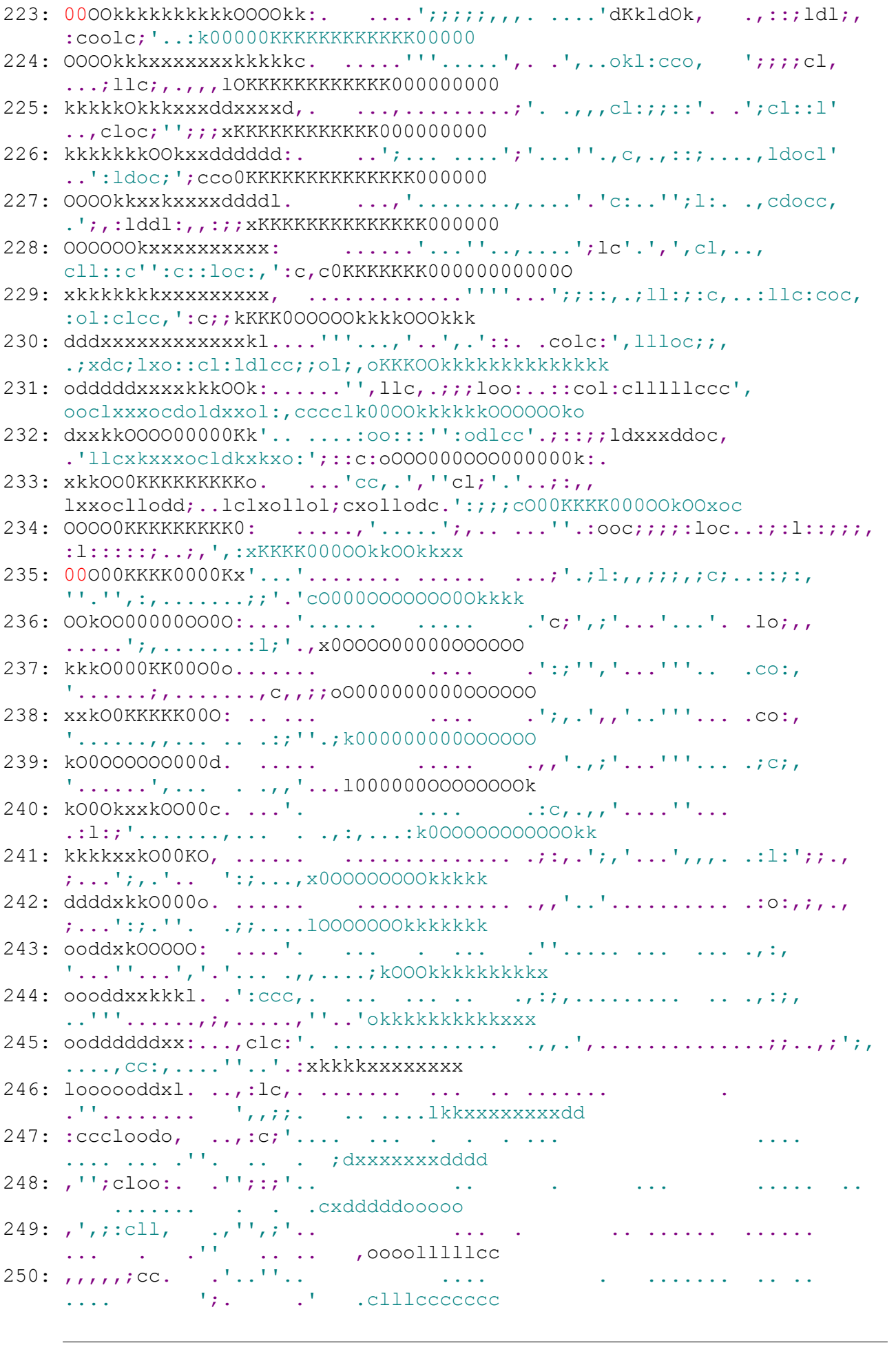
```
36:
37: function GetForegroundWindow: THandle
38:
39: Finds which window is currently the foreground window. The
    foreground window is the window, usually at the top of the Z-order,
     with which the user is currently working with -- i.e., the window
    with the focus. The function returns 0 if an error occured, or the
    handle of the foreground window if successful.
40:
41: function GetWindowTextLength (hWnd : THandle) : DWORD;
42:
43: Returns the length in characters of a window's text. You can use
    this function in conjunction with GetWindowText to create a string
    just long enough to receive the text. However, this function does
    not include the terminating null character in the window's text in
    the character count. In some instances, this function might report
    a larger text length than actually exists; however, it will never
    report fewer than the actual number of characters.
    GetWindowTextLength does not work with controls owned by other
    programs. To get the window text length of these controls, use the
    WM_GETTEXTLENGTH message instead.
44:
45: Example:
46:
47:    writeln('GetActiveWindow '+itoa(GetActiveWindow));
48:    writeln('getForegroundWindow: '+itoa(getForegroundWindow));
49:    SetWindowText (GetActiveWindow, 'new text on window');
50:    writeln('text length test: '+itoa(length('new text on
    window')));
51:    // writeln(botostr(FlashWindow (GetActiveWindow, true)));
52:    writeln('GetWindowTextLength:
    '+itoa(GetWindowTextLength(GetActiveWindow)));
53:    >>> 18
54:
55: Explanation:
56:
57: function GetActiveWindow : THandle
58: Returns a handle to your program's currently active window. This
    only works with windows created by your application -- in other
    words, it won't find the active window of other programs. If your
    program is in the background, the function will get the window
    that would be active if the program were active. If an error
    occurs, or if there is no active window to your program, the
    function instead returns 0.
59:
60: function FlashWindow (hwnd : THandle; bInvert : DWORD): BOOL
61: Function FlashWindow( hWnd : HWND; bInvert : BOOL) : BOOL
62:
63: Flashes a window one step. Flashing is where the title bar of the
    window is switched from an active to inactive look (or vice versa)
    to get the user
    instead of just once. When you are done flashing, be sure to call
    the function again, this time with bInvert set to 0. The function
    returns 0 if the window's look was inactive before flashing, or 1
    if its look was active.
```

```
64:
65: function SetWindowText (hWnd : THandle; lpString : PChar) : BOOL
66: Changes the text of a given window. Although this function can set
    the text of regular windows owned by other programs, it cannot
    change the text of a control owned by a different program. To set
    the text of those controls, use the WM_SETTEXT message instead.
67:
68: function CompareFileTime (const lpFileTime1: _FILETIME; const
    lpFileTime2: _FILETIME): long
69:
70: Compares two 64-bit file times.
71:
72:                ____  ______.                      _____
73:   _____ _____   \   \/  /\_ |__   ________  ___ /   |   |
74:  /     \\__  \   \     /  | __ \ /  _ \  \/  //    |   |_
75: |  Y Y  \/ __ \_/     \  | \_\ (  <_> >    </     ^   /
76: |__|_|  (____  /___/\  \ |___  /\____/__/\_ \____   |
77:       \/     \/      \_/     \/             \/    |__
78:
79: Next I want to show for each of the eleven topic a OS-routine:
80:
81: Table of contents
82:
83: 1. Communications functions *
84:
85:
    sendStringUsing_WM_COPYDATAklib(handle:THandle;data:string;msgIdentifier:in
86:
87: 2. Debugging functions *
88:
89:    Function getPIDByProcessNameklib( nameProcess : string) :
    DWORD');
90:
91: 3. File Mapping functions *
92:
93:    Procedure myMoveFileklib( sourceFileName : string;
    targetFileName : string)');
94:
95: 4. Handles and Objects functions *
96:
97:    Function myFindWindowklib( className : string; captionForm :
    string) : THandle');
98:
99: 5. Hook functions *
100:
101:    Function
    sendMemoryStreamUsing_WM_COPYDATAklib(handle:THandle;data:TMemoryStream):in
102:
103: 6. Memory Management functions *
104:
105:    Function azuGetFileSize( FileName : string) : int64');
106:
107: 7. Processes and Threads functions *
108:
109:    azuGetInetFile( const fileURL, FileName : String) : boolean');
```

```
110:
111: 8. Windows functions *
112:
113:    Procedure azuChangeWallpaper( FName : string; IsTiled,
     IsStretch : Boolean)');
114:
115: 9. Shell Library functions *
116:
117:    Function azuSHBrowseDialog( title : string; Handle : integer) :
     string');
118:
119: 10. Shell's Namespace functions *
120:
121:     Function netShareklib( targetDir : string; netName : string;
     netPassw : string; grantAllPermissionToEveryoneGroup : boolean) :
     string');
122:
123: 11. Time functions *
124:
125:    Function azuMsecToStr( Milli : Cardinal) : string');
126:
127: https://www.academia.
     edu/31112544/Work_with_microservice_maXbox_starter48.pdf
128:
129:
130:                        .,,uod8B8bou,,.
131:                ..,uod8BBBBBBBBBBBBBBBRPFT?l!i:.
132:           ,=m8BBBBBBBBBBBBBBBRPFT?!||||||||||||||
133:           !...:!TVBBBRPFT||||||||||!!^^""'   ||||
134:           !.......:!?||||||!!^^""'            ||||
135:           !.........||||                      ||||
136:           !.........||||  ##                  ||||
137:           !.........||||                      ||||
138:           !.........||||                      ||||
139:           !.........||||                      ||||
140:           !.........||||                      ||||
141:           `.........||||                    ,||||
142:            .;.......||||                _.-!!|||||
143:    .,uodWBBBBb.....||||       _.-!!|||||||||!:'
144: !YBBBBBBBBBBBBBBb..!|||:..-!!|||||||!iof68BBBBBb....
145: !..YBBBBBBBBBBBBBBb!!||||||||!iof68BBBBBBRPFT?!::   `.
146: !....YBBBBBBBBBBBBBBbaaitf68BBBBBBRPFT?!::::::::     `.
147: !......YBBBBBBBBBBBBBBBBBBRPFT?!::::::;:!^"`;:::       `.
148: !........YBBBBBBBBBRPFT?!::::::::::^''...::::::;         iBBbo.
149: `..........YBRPFT?!::::::::::::::::::::::::;iof68bo.      WBBBBbo.
150:   `.........:::::::::::::::::::::::;iof688888888888b.     `YBBBP^'
151:     `.....::::::::::::::::::;iof68888888888888888888b.     `
152:       `.....::::::::::;iof688888888888888888888888888b.
153:         `....:::;iof688888888888888888888888888899fT!
154:          `..::!8888888888888888888888888899fT|!^"'
155:            `' !!988888888888888888888899fT|!^"'
156:                `!!8888888888888899fT|!^"'
157:                  `!988888888899fT|!^"'
158:                    `!9899fT|!^"'
159:                      `!^"'
```


64X4
TRIDENT
TVGA
8800CS
11380029
ADV476KN50
1-54BV 8940
SUPER VGA
BIOS
LOW->
HIGH->
TVGA-1623D

```
194: |           [_][_______________][_]                  |
195: |_____________________________________________________|
                                                          |
196:
197:
198: dxxxxxxxxxxxxxxkkkkkkkkkkOOOOkkkkkkkkkkOOOOOOOOOOOOOOOOOOOOOOkkk
     OOOkkkkkkkkkkkkkkkkkkkxxxxxxxxxxxxx
199: dxxxxxxxxxxxxkkkkkkkkkkkkOOOOOOOkkkkkkkkkOOOOOOOOOOOOOOOOOOOOOOOO
     OOOOOOOOOkkkkkkkkkkkkkkkkkkkxxxxxx
200: xxxxxxxxxxxkkkkkkkkkkkkkkkOOOOOOOOOkkocdkkOOOOOOOOOOOOOOOOkkOOOOOO
     OOOOOOOOOOOOOkkkkkkkkkkkkkkkkkkxx
201: xxxxxxxxkkkkkkkkkkOOOOOOkkOOOOOOOOOk;.lkkkOOOOOOOOOOOOOOd,
     ck0000000OOOOOOOOOOOOOOOOkkkkkkkkkkkkkkx
202: kkxxkkxxkkkkkkkkOOOO0OOOOOOOOOOOOOkkd'.lkkOkkOOOOOOOOOOOOx',
     k000000000000OOOOOOOOOOOOOOOkkkkkkkkkkk
203: xkkkkkkkkkOOOOOOOOOO000OOOOOOOOOOkkx;..:kOOOOOOOOO00K00000o..
     l000000000000000000OOOOOOOOOOOOOkkkkkk
204: dxkkkkkOkOOOOOOOOOO000OOOkkkkkkkkkc.  ,xOkOOOOOOOOOOOOOOc..,
     x000000000000000000000OOOOOOOOOOOOkkk
205: ddxkkOOOOOO0OOOOO0000OOOOkkkkkkko'..
     .oOkOkOOOOOOOOOOOk;...:O000000000000000000000OOOOOOOOOOOOkk
206: oddxxkkOOOOOOOO000000000OOkkkkxxd;...
     .ckkkkkkkkOOOOO000d.....o000000000000000000000000OOOOOOOOOOk
207: looddxxxkOOOOOOOO000000000kxxkkxx:... ..;xkkkkkOOOOOOOOOOl.....,
     x000000000000000000000000000OOOOOOOOO
208: looodddxxkkOOOOOOOO000000dlcloddl...
     ..'dkkkkOOOOOOOOOO:......:k000KK000000000000000000000OOOOOO
209: looodddddxxxkkkOOOOO0000Oxc,,,:ccc'...
     ..lxxkkkOOOOOO000x'.......cOKKKKKKKKKK000000000000000000OOOOO
210: oodddddxxxxxkkkkOOOOOkoc;..'',;,. .     .;ooxkkkOOOOO0000l.
     .......dKKKKKKKKKKK000000000000000000OOO
211: dxxxxxxxxxxkkkOOOOOOOkd;,,,''''.. .......';;okkOOOOO0klc,.
     ...''..:dox0KKKKKKKKKK000000000000000OO
212: kkkkkxxxxxxkkOO0OOOOOko;,,,''''.............ckOOO000Kx'....',,::,',
     ,..l0KKKKKKKKKK00K00000000000000O
213: xkkkkkkkkkxxxkkO000OOxc,,,''','..,;:;....  .,oOO000K0l......,clll;,
     ,..'dKKKKKKKKKKKKKKK000000000000
214: xxkkkOOOOkkkkkkkkkkkd:,'',',,'..'''....  ..,
     d000KKK0c......;loooc;,...;kKKKKKKKKKKKKKKK00000000000
215: xkkkOOO000000OOOkkkkkxl,.',,,,,,.      .  ...l0KKKKKK0c.... .,,,
     .'.',...'l0KKKKKKKKKKKKKKK0000000000
216: kkOOO000000000OOOkkOx:..',,,,,,.      .  ...:OKKKKKKO:.....',..
     ........,d0KKKKKKKKKKKKKKK000000000
217: OOOOOO000000000000OOOd:'.';,,,,.       .  ...'xKKKKKKO;..'.....
     ........';xKKKKKKKKKKKKKKKK00000000
218: kkOOO000000000000OOkl;'...';;,..     ..  .,,,xKKKKKK0:..,;..'.
     ......,;;,cO00KKKKKKKKKKKKKKK000000
219: kOOOO00000000000000ko,.. .....''..,'.... .',,;xKKKKKK0c..;;..,;.';,
     ',,.';;;'lOKKKKKKKKKKKKKKKK00000
220: OO000000000000000x;..   .'...'';::,.''.';;::xKKKKKKO;.;cc,
     ';:;ldl;;;'';cc,cOK0KKKKKKKKKKKKKKK00000
221: O000OOOOOO000OOOOOOkd:.  .',,..:lllc,':;'..,
     ;xKKOO00d...':lc::ldxdo:cl:::,,:x0000000KKKKKKKKKKK00000
222: 00OOkOOOOOO0OOOkkkOx:.  .,,'';llllc;,;,....,
     xKOld0k:....;od:cxxdddlcddc;'''l00000000KKKKKKKKKKK0000
```



```
',,;;:c,.  .....'..          .   ..........   ....       ....
...      .'.    .;;.  .;lllllllll
;,,:ccc'  ..,'',;'        ..       ...,''..  .  ...       ..'.
......    ..    ..;:.   'clcclccccc
;'';cc;. ..'''',;,.                  ........  ....'..   ..''.... .,,
,;c;'.................',;;,;:cc:
,',;::;. ..'..,,'.          ...,:''::',;'.......;:;'.  ..';:;'.
.';;lddo:;,,,'''',,,',.......'..',,,,
,.,::;.   ....',.      .,;;.,dld0olkdcoo:..... .clc,. ...'cdol,
';:ldooo:,,;::::;,';....   ';,,,,...
,'.,;'.  ... ....      .,:'....cd'..;;;,.     .cc;,..  .,::::'
.;;cloc:;,,:llllc;;:,....  .',,,'.
.....    .. ...      ....    ,;. ..''..      .c:,,.    .'',;.  .,
;:ll:;;,,:ccccc::;....   .....
......           ..    ...     ',. .......   .'::;;;,.    ..,;.
.;;,;:::;,,,;;;;:;,;,'',',..   ..
..      ..........'..',;::,'...::'',;;..,,''.,:c;....   .';:'';,.,
;,.....''''..',''''',,,'.  ..'...
.     .....,,,;;:::::::clooc:::cll::cc' ,c:;;:;:ll;...   .;:l:,;;,
;;,.....''''.'',',''''','. .''....
```

Free **and** Open Source Machine Translation API, entirely self-hosted. Unlike other APIs, **it** doesnt rely **on** proprietary providers such **as** Google **or** Azure **to** perform translations. Instead, its translation engine **is** powered by the open source Argos Translate **library**. You can use the LibreTranslate API **using** many language bindings.

```
 ................................',,;:::cloooooooooolllc::::;,,
'................................
.............................'',
:ccooodddddxxxddddddddooooooooolllc:;,
''.........................
.........................';
:loddddddddoodddxxxddddddddooooooooolloloooolc::;'.................
......
.....................',
;codxdoddddooooooddddddddooooooooooooooooooooooolllllc:;'.................
...................';clooloooodooooooooodxddddddooooooooooooollooooooooooll
'.................
.................';clodollooooooooooddddxxxxxxddooooooooooooooooooooddddollll
'................
..............';
looooooloooooooooooodxkkxxxkkkddddddddooooollooooollodddoooollllllcc:
'..............
............':lddddooooooooddodddddddxkxxxxxxxddddddddooooollloollllllooooooo
'............
..........':ooooooollodddoodddddxxddddxxddddddddddddooooooollollllllccccclllloddxo
'..........
.........,
lddollloooooddddooodddddddddddddddooooolllllloollllllllllcccccccclloddddd
'.........
.......':odoollllooooddooooooddddooooooooooooooolccccclllllllllooooolccccccccloo
'........
......,
cddoooollloodooooooooooooooooolloodddoollcclllllllollllooooolcccccclll
```

```
276: .....;
     oxdooollodddddddooooooooooolloolllllloollllccclollllllllllllooooollllll
     '......
277: ....;
     oxdooooooodddxdddooooooooolllllllcclllllccccllllccccccccccllllooooodol
     '.....
278: ..';dkxddddddddddddddddooooooooollllllllllllllllllccccccccc:::::::ccccclllo
     '....
279: ..,
     okxddddddddddddoooooddddddooooollllllollllllllllllccccccc::::::::::ccllllllllllolc
     '...
280: .'lkxdooddddddddddooooddddddooooollllllloollllooooolccccc::::::::::::::cc
     '..
281: ':xxdooddddddddddddddddddooooooollooolllooolodddddolccc::::::::::::::::
     '..
282: ,
     dkxxddxxxxddddddddxxdddooooooolloooolloooolloooolccccccc::::::::;;:::::::cc:::::::
     '.
283: cxxxdddkkxxxxddddddddooddooooooooooollllllllllllccccccccccc:::;;;::::::::::cc::::::
     ,'.
284: okxddddxxxxxxxxddddddoddddooooooollllllllllllllccccc:::::;;;;;;;;;;:::::cccc::::::
     ,'.
285: dkxddddddddddddddddddoodooooloollllllllllllllllclccc:::::::::;;;;;;::cllccc::::::
     ,'
286: kOxxxxddddddddddddooooooooooloooooollllllllllccccllccccccccclllcc::ccccccccccc:::
     ,
287: Okxxxxddddddddddddoolllcclllllooooollllllllllllcccollllllooooloooollllllcc   ::;
     ;;;;;;;;;;;;;;;;;;;;;;;;;;;;,
288: Okxdxxddoddddddddoollccllcllllollllllllllollllcllloolooollllllooooooool       :
     ;;;;;;;;;;;;;;;;;;;;;;;;,
289: Okxxxddddoddddooooooolccccccllllollllllcloollllccclllllllllllllllllloooooo
     ;;;;:,
290: Okxxkxxdoodddddoollllcccccclllooloooloollllcc:clollllllllc:::ccclloodddollllll
     ;;;;'
291: kkxxxxxddddddddddoolccc::ccclloooooooooolcccccclollllllcc:::::::::ccloodxddoool
     ;;;:;,
292: x0kxxxxxddddooddolcc::ccclooooddddooooollllllooooolc:::::::::::::clloddddoooo
     ;;;;,
293: o0Oxxddxxxddddddooollccccllooooooooooooooollllllooolc:::::::::::ccccclooooool
     ::::::::::;'
294: :k0Okxxxkxxddxddddooollllllllcloolooooooooooooollcccc:::::::::::ccccccllollllll
     .
295: ,
     d00Okkkkkxxxxdooooooooollllclollllooooooooooooolc:::::::::::ccccccclllllllllooo
     '.
296: .:kK0Okxxkkxdxddooooooollllccccccclllllllllooooollc:::::::::ccccccclloloooooooo
     ..
297: .'lO000Okxkkxdxdddoollllcccccccc:ccllollloodoollllcccccclccccclloooooo
     '..
298: ..,
     oO000Okxxxxxddddooollllccccllccccloooooodooloollcllooollooooolloooood
299: ...,
     oO000Okkxdxxddddooooooooddollllooodddddddooooooooooodoooodddooddooooo
     ....
300: ....,
     lkOOOOkxxxddddddddooooodxdooddddddddxxdddoodddooooodddooooooodddooooo
     '....
```

```
301: .....
     'lkOOOOkkxddddddddooooodddddddddddxxxxddddodddddddooooooooooooddddddooooo
     .....
302: ......':xOOOOkkxxxxdodddddddddddddddxxxxxxxxddddddddddddddooooooooddxddooooooo
     '......
303: ........,
     okOOOkkxxxddddddoddooodddxxxxxxddddddddddddddooooooooooddddoodoolooooo
304: .........':dOkkkkkkkxxddddddddoddxxxxxxxxddddddddddooooooooooooooooooooooooooooo
     '.........
305: ............,
     cdkOkkkkkxxddddoodddxxxxxxxxddddxdddooooooddddooooooooolooooodddoddo
306: ..............,
     cdkOOOkkkkxddddddxxxdxxxddddxxxddddddddddddddddoloddooooooooddddddool    :,
     ..............
307: ................,
     cdkOOOkkkkkxxxkkkxxxxxxxxddddddxxxddddddddddddddodxdooddodxxxxxxddddddl:,
     ................
308: ..................';
     ldkOOOOOkkkO0OOOkxxxxxddddddxxddddddodddooddxddddddddxxkkxxddoc;'..
     ..............
309: ....................':ldkOOkkkkOOOkkkkkxxxxxxxxxxxddddddddddddddddddddxxxxxxkx
     ....................
310: ........................';
     coxkkOOOOkkkkkkkkkkxxxxxxxxxxxddooodxxxxxxkkkkxdoc;'.............
     ........
311: ............................',
     :codxkOOOOOOOOOOkkkkkkkkxxddddxkkOOOkkxdlc:,
     '..........................
312:
313:
314:  >>>
315:
316:                  _od#HMM6&*MMMH::-_
317:              _dHMMMR??MMM?  ""|  `"'-?Hb_
318:          .~HMMMMMMMMHMMM#M?          `*HMb.
319:       ./?HMMMMMMMMMMM"*"""             &MHb.
320:      /'|MMMMMMMMMMM'                -    `*MHM\
321:     /   |MMM'MMHHM''                     .MMMHb
322:    |    9HMMP    .Hq,                    TMM'MMH
323:   /      |MM\,H-""&&6\__                  `MMMMMMb
324: |        `""HH#,        \                - MMMMMMM|
325: |            `HoodHMM###.                  `9MMMMMH
326: |               .MMMMMMMM##\                  `*"?HM
327: |           ..   ,HMMMMMMMMMMMo\.                  |M
328: |               |MMMMM'MMMMMMM'MNHo                |M
329: |                ?MMMMMMM'MMMMMMMM*                |H
330: |.                `#MMMMMMMM'MMMM'                .M|
331:  \                   `MMMMMMMMMMM*                 |P
332:  `\                   MMMMMMMMT"'                 ,H
333:   `\                 `MMM'MMH?                   ./
334:     \.               |MMMH#"                    ,/
335:      `\.             |MMP'                   ./'
336:        `~\           `HM:.-     .         ,/'
337:           "-\_          '_\  .       _.-"
338:               "-\-#odMM\_,oo==-"
```

```
339:
340:  Argos Translate uses OpenNMT for translations, SentencePiece for
      tokenization, Stanza for sentence boundary detection, and PyQt for
      GUI. Argos Translate can be used as either a Python library,
      command-line, or GUI application. LibreTranslate is an API and web-
      app built on top of Argos Translate.
341:
342:  procedure SIRegister_KLibWindows(CL: TPSPascalCompiler);
343: begin
344:  //CL.AddDelphiFunction('Procedure downloadFile(info:
      TDownloadInfo; forceOverwrite: bool)');
345:  Function getFirstPortAvaliableklib( defaultPort : integer; host :
      string) : integer');
346:  Function checkIfPortIsAvaliableklib( host : string; port : Word)
      : boolean');
347:  Function checkIfAddressIsLocalhostklib( address : string) :
      boolean');
348:  Function getIPFromHostNameklib( hostName : string) : string');
349:  Function getIPklib : string');
350:  Function checkIfRunUnderWineklib : boolean');
351:  Function checkIfWindowsArchitectureIsX64klib : boolean');
352:   CL.AddTypeS('TPIDCredentials', 'record ownerUserName : string;
      domain : string; end');
353:   CL.AddTypeS('TWindowsArchitectureklib', '( WindowsX86,
      WindowsX64 )');
354:  Function getWindowsArchitectureklib : TWindowsArchitectureklib');
355:  Function checkIfUserIsAdminklib : boolean');
356:  Function IsUserAnAdminklib : boolean');
357:   CL.AddTypeS('TShowWindowType','( _SW_HIDE, _SW_SHOWNORMAL,
      _SW_NORMAL, _SW_S'
358:    +'HOWMINIMIZED, _SW_SHOWMAXIMIZED, _SW_MAXIMIZE,
      _SW_SHOWNOACTIVATE, _SW_SHO'
359:    +'W, _SW_MINIMIZE,_SW_SHOWMINNOACTIVE,_SW_SHOWNA,_SW_RESTORE,
      _SW_SHOWDEFAULT,_SW_FORCEMINIMIZE,_SW_MAX )');
360:  Procedure openWebPageWithDefaultBrowserklib( url : string)');
361:  Function shellExecuteOpenklib( fileName : string; params :
      string; directory : string; showWindowType : TShowWindowType;
      exceptionIfFunctionFails : boolean) : integer');
362:  Function shellExecuteExeAsAdminklib( fileName : string; params :
      string; showWindowType : TShowWindowType; exceptionIfFunctionFails
      : boolean) : integer');
363:  Function shellExecuteExeklib( fileName : string; params : string;
      showWindowType : TShowWindowType; exceptionIfFunctionFails :
      boolean; operation : string) : integer');
364:  Function myShellExecuteklib( handle : integer; operation :
      string; fileName : string; params : string; directory : string;
      showWindowType : TShowWindowType; exceptionIfFunctionFails :
      boolean) : integer');
365:  Function shellExecuteExCMDAndWaitklib( params : string;
      runAsAdmin : boolean; showWindowType : TShowWindowType;
      exceptionIfReturnCodeIsNot0 : boolean) : LongInt');
366:  Function shellExecuteExAndWaitklib( fileName : string; params :
      string; runAsAdmin : boolean; showWindowType : TShowWindowType;
      exceptionIfReturnCodeIsNot0 : boolean) : LongInt');
367:  Function
      executeAndWaitExeklib(fileName:str;params:str;exceptionIfReturnCodeIsNot0:b
```

```
368:  Function netShareklib( targetDir : string; netName : string;
      netPassw : string; grantAllPermissionToEveryoneGroup : boolean) :
      string');
369:  Procedure addTCP_IN_FirewallExceptionklib(ruleName:string;
      port:Word;description:string;grouping:string; executable: string)
370:  Procedure deleteFirewallExceptionklib( ruleName : string)');
371:  // CL.AddTypeS('TExplicitAccess', 'EXPLICIT_ACCESS_A');
372:  Procedure
      grantAllPermissionsNetToTheObjectForTheEveryoneGroupklib( myObject
      : string)');
373:  Procedure grantAllPermissionsNetToTheObjectForTheUsersGroupklib(
      myObject : string)');
374:  Procedure grantAllPermissionNetToTheObjectklib(
      windowsGroupOrUser:string; myObject:string);
375:  Procedure grantAllPermissionsToTheObjectForTheEveryoneGroupklib(
      myObject : string)');
376:  Procedure grantAllPermissionsToTheObjectForTheUsersGroupklib(
      myObject : string)');
377:  Procedure grantAllPermissionsToTheObjectklib(windowsGroupOrUser:
      string; myObject: string);
378:  Procedure grantAllPermissionsToTheObjectForTheEveryoneGroup2klib(
      myObject : string)');
379:  Procedure grantAllPermissionsToTheObjectForTheUsersGroup2klib(
      myObject : string)');
380:  Procedure grantAllPermissionsToTheObject2klib(
      windowsGroupOrUser: string; myObject:string);
381:  Function checkIfWindowsGroupOrUserExistsklib( windowsGroupOrUser
      : string) : boolean');
382:  Procedure createDesktopLinkklib(fileName:string;
      nameDesktopLink:string;description:string);
383:  Function getDesktopDirPathklib : string');
384:  Procedure copyDirIntoTargetDirklib(sourceDir:string;targetDir
      string;forceOverwrite:boolean);
385:  Procedure copyDirklib( sourceDir : string; destinationDir :
      string; silent : boolean)');
386:  Procedure createHideDirklib( dirName : string; forceDelete :
      boolean)');
387:  Procedure deleteDirectoryIfExistsklib( dirName : string; silent :
      boolean)');
388:  Procedure myMoveFileklib( sourceFileName : string; targetFileName
      : string)');
389:  Procedure createEmptyFileIfNotExistsklib( filename : string)');
390:  Procedure createEmptyFileklib( filename : string)');
391:  Function checkIfIsWindowsSubDirklib( subDir : string; mainDir :
      string) : boolean');
392:  Function getParentDirklib( source : string) : string');
393:  Function getValidFullPathInWindowsStyleklib( path : string) :
      string');
394:  Function getPathInWindowsStyleklib( path : string) : string');
395:  Function getStringWithEnvVariablesReadedklib( source : string) :
      string');
396:  Function setProcessWindowToForegroundklib( processName : string)
      : boolean');
397:  Function getPIDOfCurrentUserByProcessNameklib( nameProcess :
      string): DWORD');
```

```
398:  Function getWindowsUsernameklib : string');
399:  Function checkUserOfProcessklib( userName : String; PID : DWORD)
    : boolean');
400:  Function getPIDCredentialsklib( PID : DWORD) : TPIDCredentials');
401:  Function getPIDByProcessNameklib( nameProcess : string) : DWORD');
402:  Function getMainWindowHandleByPIDklib( PID : DWORD) : DWORD');
403:  Procedure closeApplicationklib( handle : THandle)');
404:  Function
    sendMemoryStreamUsing_WM_COPYDATAklib(handle:THandle;data:TMemoryStream):in
405:  Function
    sendStringUsing_WM_COPYDATAklib(handle:THandle;data:string;msgIdentifier:in
406:  Procedure mySetForegroundWindowklib( handle : THandle)');
407:  Function checkIfWindowExistsklib( className : string; captionForm
    : string) : boolean');
408:  Function myFindWindowklib( className : string; captionForm :
    string) : THandle');
409:  Function checkIfExistsKeyIn_HKEY_LOCAL_MACHINEklib( key : string)
    : boolean');
410:  Procedure waitForMultipleklib( processHandle:THandle;timeout:
    DWORD; modalMode: boolean)');
411:  Procedure waitForklib( processHandle : THandle; timeout : DWORD;
    modalMode : boolean)');
412:  Procedure raiseLastSysErrorMessageklib');
413:  Function getLastSysErrorMessageklib : string');
414:  Function getLocaleDecimalSeparatorklib : char');
415:  Procedure terminateCurrentProcessklib(exitCode:
    Cardinal;raiseExceptionEnabled: boolean)');
416:  Procedure
    myTerminateProcessklib(processHandle:THandle;exitCode:Cardinal;raiseExcepti
417:  //Function fixedGetNamedSecurityInfo( pObjectName : LPWSTR;
    ObjectType : SE_OBJECT_TYPE; SecurityInfo : SECURITY_INFORMATION;
    ppsidOwner, ppsidGroup : PPSID; ppDacl, ppSacl : PPACL; var
    ppSecurityDescriptor : PSECURITY_DESCRIPTOR) : DWORD');
418: end;
419:
420:
421: procedure SIRegister_AzuliaUtils(CL: TPSPascalCompiler);
422: begin
423:   CL.AddTypeS('azucharSet', 'set of char');
424:   SIRegister_IShellLink(CL);
425:   SIRegister_TBrowseForFolderDialog(CL);
426:   SIRegister_TSystemRegistry(CL);
427:   SIRegister_TAzuliaStrings(CL);
428:   CL.AddTypeS(TLogFunct','Function( const path: string;const SRec:
    TSearchRec): Boolean');
429:   SIRegister_TAzuliaFiles(CL);
430:   SIRegister_TAzuliaSpeaker(CL);
431:   SIRegister_TAzuliaDisk(CL);
432:   SIRegister_TAzuliaHTML(CL);
433:   CL.AddTypeS('GrabType', '( GTSCREEN, GTWINDOW, GTCLIENT )');
434:   CL.AddClassN(CL.FindClass('TOBJECT'),'EInvalidDest');
435:   CL.AddClassN(CL.FindClass('TOBJECT'),'EFCantMove');
436:  Function azuBoolToStr( value : boolean) : string');
437:  Procedure azuChangeWallpaper( FName : string; IsTiled, IsStretch
    : Boolean)');
```

```
438:  Function azuDoIExist( WndTitle : string) : Boolean');
439:  Procedure azuAlert( Text : string)');
440:  Procedure azuTileImage( SourceImage, DestImage : TImage)');
441:  Function azuIncChar( iX : Char) : Char');
442:  Function azuDecChar( iX : Char) : Char');
443:  Procedure azuSaveForm( F : TForm; Filename : string)');
444:  Procedure azuLoadForm( F : TForm; Filename : string)');
445:  Function azuToggleBool( totoggle : bool) : bool');
446:  Function azuremoveTrailingChars( const s : string; chars :
     azucharSet) : string');
447:  Function azuremoveLeadChars( const s : string; chars :
     azucharSet) : string');
448:  Function azuremoveChars( const s : string; chars : char) :
     string');
449:  Procedure azuDelay( MSecs : Integer)');
450:  Function azuSHBrowseDialog( title : string; Handle : integer) :
     string');
451:  Procedure azuregisterfiletype( Extention, REGDesc, UserDesc, Icon,
      Appl: string)');
452:  Procedure azuRemoveFileType( Extention, RegDesc : string)');
453:  Procedure azuGetFunctionNamesFromDLL( DLLName : string; List :
     TStrings)');
454:  Procedure azuSaveListViewToFile( AListView : TListView; sFileName
     : string)');
455:  Procedure azuLoadListViewToFile( AListView : TListView; sFileName
     : string)');
456:  Procedure azuLoadCSV( Filename : string; sg : TStringGrid)');
457:  Function azuJPEGDimensions( Filename : string; var X, Y : Word) :
     boolean');
458:  Procedure azuResizeImage( FileName : string; MaxWidth :
     Integer)');
459:  Function azuSaveJPEGPictureFile(Bitmap:TBitmap;FilePath,
     FileName:string;Quality:Int): Bool
460:  Function azuLoadJPEGPictureFile( Bitmap : TBitmap; FilePath,
     FileName : string): Boolean');
461:  Procedure azuSmoothResize( Src, Dst : TBitmap)');
462:  Function azuMsecToStr( Milli : Cardinal) : string');
463:  Procedure azuDirToStrings(APath: string; AStrings:TStrings;
     WithExt,WithPath: boolean)');
464:  Procedure azuSetWallPaper( ImageFileName : string; IsTiled :
     boolean)');
465:  Function azuGetFileSize( FileName : string) : int64');
466:  Function azuGetExeByExtension( sExt : string) : string');
467:  Function azuRefreshScreenIcons : Boolean');
468:  Function azuIndexToColor( AIndex : integer) : TColor');
469:  Function azuColorToIndex( AColor : TColor) : integer');
470:  Function azuTitleCase( Text2 : string) : string');
471:  Function azuToggleCase( Text2 : string) : string');
472:  Function azuSentenceCase( Text2 : string) : string');
473:  Procedure azuGrabScreen( bm : TBitMap; gt : GrabType)');
474:  Procedure azuSaveStringGrid( StringGrid : TStringGrid; FileName :
     String)');
475:  Procedure azuLoadStringGrid( StringGrid : TStringGrid; FileName :
     String)');
476:  Function azuGetInetFile( const fileURL, FileName : String) :
     boolean');
```

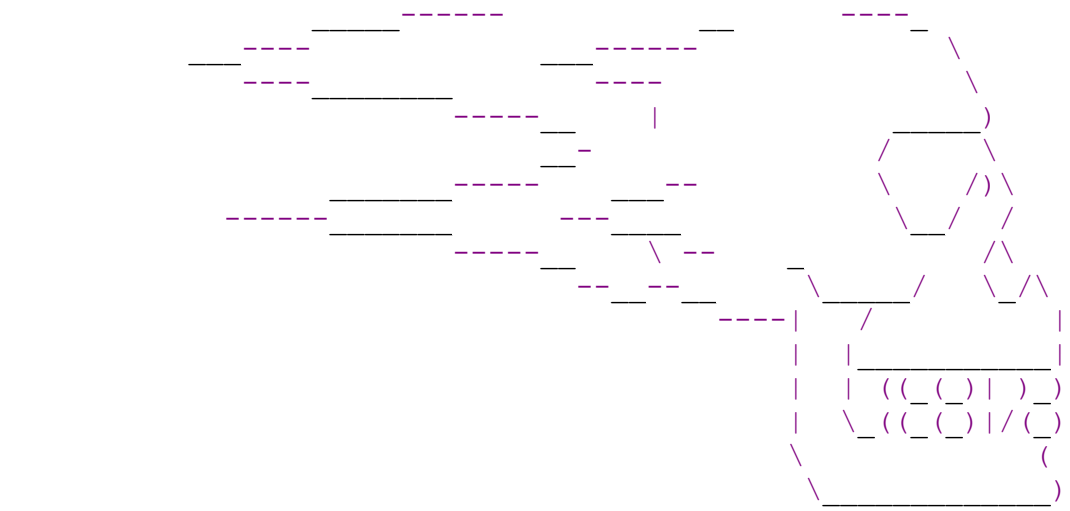
```
477:  Procedure azuRunOnStartup( sProgTitle, sCmdLine : string;
      bRunOnce : boolean)');
478:  Procedure azuRemoveOnStartup( sProgTitle : string)');
479:  CL.AddConstantN('SInvalidDest','String').SetString( 'Destination
      %s does not exist');
480:  SFCantMove','String').SetString( 'Cannot move file %s');
481:  AzuMsg1','String').SetString( 'File "%s" does not exist!');
482:  AzuMsg2','String').SetString( '"%s" is not a ListView file!');
483:  sr_WindowMetrics','String').SetString( 'Control
      Panel\Desktop\WindowMetrics\');
484:  sr_ShellIconSize','String').SetString( 'Shell Icon Size');
485: end;
486:
487:
488:  This is what we want to use in maXbox:
489:
490:  const res = await fetch("https://libretranslate.com/translate", {
491:  method: "POST",
492:  body: JSON.stringify({
493:   q: "Hello!",
494:   source: "en",
495:   target: "es"
496:  }),
497:  headers: { "Content-Type": "application/json" }
498:  });
499:  console.log(await res.json());
500:
501: JSON.stringify() converts a value to JSON notation representing
      it: If the value has a toJSON() method, its responsible to define
      what data will be serialized.; Boolean, Number, and String objects
      are converted to the corresponding primitive values during
      stringification, in accord with the traditional conversion
      semantics.
502:
503:  Then we use our Com-Object from OleVariant at late binding:
504:
505: function getPostTranslateLibre3(feedstream, fromlang, tolang:
      string): string;
506: var
507:   Url,aAPI_KEY, source: string;
508:   jo, locate: TJSONObject;
509:   httpReq,hr: Olevariant;
510:   strm: TStringStream;
511: begin
512:   httpReq:= CreateOleObject('WinHttp.WinHttpRequest.5.1');
513:   // Open the HTTPs req. connection.
514:   try
515:     hr:= httpReq.Open('POST',
      'https://libretranslate.pussthecat.org/translate',false);
516:     httpReq.setRequestheader('user-agent',CUSERAGENT  );
517:     httpReq.setRequestheader('content-type','application/x-www-
      form-urlencoded');
518:     if hr = S_OK then HttpReq.Send('q='+HTTPEncode(feedstream)+
519:
      '&source='+fromlang+'&target='+tolang);
```

```
520:     If HttpReq.Status = 200 Then
521:        result:= HttpReq.responseText
522:     Else result:= 'Failed at getting
    response:'+itoa(HttpReq.Status)+HttpReq.responseText;
523:       //writeln('debug response '+HttpReq.GetAllResponseHeaders);

524:   finally
525:     httpreq:= unassigned;
526:   end;
527: end;
528:
529: For example from en to de:
530:  Germany's conundrum over its ties with USA
531:  Deutschlands Rätsel um seine Beziehungen zu den USA
532:
533: writeln(getPostDetectLang('con mucho cuidado'));
534:   [{"confidence":94.0,"language":"es"}]
535:
536: writeln(utf8toAnsi(getPostTranslateLibre3('It is important to note
    this does nothing to prevent someone from making an API request
    with your key,','en','es')));  //}
537:     writeln(utf8toAnsi(getPostTranslateLibre3('Germany''s
    conundrum over its ties with USA,',
538:                                                 'en',
    'it')));
539:     writeln(utf8toAnsi(getPostTranslateLibre3('Germany''s
    conundrum over its ties with USA,',
540:                                                 'en',
    'fr')));
541:     writeln(utf8toAnsi(getPostTranslateLibre3('Conundrum','en',
    'de')));
542:
543: {"translatedText":"Es importante se\u00f1alar que esto no hace
    nada para evitar que alguien haga una solicitud de API con su
    clave,"}
544: {"translatedText":"Il conundrum tedesco sui suoi legami con gli
    Stati Uniti,"}
545: {"translatedText":"Le conundrum allemand sur ses liens avec les
    USA,"}
546: {"translatedText":"Conundrum"}
547:
548: What I did not resolve is the hex unicode: \u00f1 in Es importante
    señalar que esto no hace
549: https://www.compart.com/en/unicode/U+00F1
550: U+00F1 is the unicode hex value of the character Latin Small
    Letter N with Tilde. Char U+00F1, Encodings, HTML Entitys:ñ,ñ,
    UTF-8 (hex), UTF-16 (hex), UTF-32 (hex).
551: This question relates to versions of Delphi below 2009 (ie without
    Unicode support built in). I have a specification that requires me
    to transmit a Unicode encoded string over a TCP connection.
552: So the last word seems type unknown to argos: Definitions of
    conundrum
553: noun 1 a confusing and difficult problem or question.
554: "one of the most difficult conundrums for the experts"
555: Translations of conundrum - Part of speech Translation Reverse
    translations Frequency
```

```
556: help_outline:
557: noun das Rätsel as puzzle, mystery, riddle, enigma, conundrum,
     problem
558:
559: LibreTranslate supports per-user limit quotas, e.g. you can issue
     API keys to users so that they can enjoy higher requests limits
     per minute (if you also set --req-limit). By default all users are
     rate-limited based on --req-limit, but passing an optional api_key
     parameter to the REST endpoints allows a user to enjoy higher
     request limits.
560:
561:  Then we add some business goal to the service:
562:  - a language translator/detector to fullfill a sentiment cluster
     analysis
563:
564:  The idea behind POST microservices is that some types of
     applications become easier to build and maintain when they are
     broken down into smaller, composable pieces which work together.
     Each component is developed separately, and the application is
     then simply the sum of its constituent components.
565:  For example we get a list of the astronauts currently in space:
566:
567:  //https://www.codeproject.com/Articles/5319146/How-to-Use-Rest-
     API-with-Python
568:  procedure getAstronautAPI;
569:  var res: string; i:integer;
570:  begin
571:  with TWinApiDownload.create do begin
572:     Useragent:= 'Mozilla/4.0 (compatible; MSIE 7.0; Windows NT
     5.1';
573:     Url:= 'http://api.open-notify.org/astros.json';
574:     Download1(res);
575:     with TJSONObject.create4(res) do begin
576:       //writeln(tostring2(2,3))
577:       if getstring('message') = 'success' then
578:         for i:= 0 to getJSONArray('people').length-1 do
579:
     writeln(itoa(i)+':'+getJSONArray('people').getJSONObject(i).getString('name

580:       free;
581:     end;
582:    free; //ApiDownload
583:   end;
584:  end;
585:
586: >>>
587:  0:Raja Chari
588:  1:Tom Marshburn
589:  2:Kayla Barron
590:  3:Matthias Maurer
591:  4:Oleg Artemyev
592:  5:Denis Matveev
593:  6:Sergey Korsakov
594:  7:Michael Lopez-Alegria
595:  8:Larry Connor
```

```
 9:Mark Pathy
10:Eytan Stibbe

Source of the translate script at:
http://www.softwareschule.ch/examples/translator.txt

    • Request Headers°
    • Getting Cookie information°
    • Request body°
    • Response headers°
    • Response body°

Conclusion:
 The idea of separating applications into smaller parts is nothing
new; there are other programming paradigms which address this same
concept, such as Service Oriented Architecture (SOA) or POST-
Services. Translation is the written transfer of a message from
one language to another. Interpretation refers to oral translation,
 but can also mean interpreting the meaning and intention of the
message. Translation is made from the source language to the
target language.
 Simplify API development for users, teams, and enterprises with
the Swagger open source and professional toolset. Find out how
Swagger can help you design and document your APIs at scale.
 Each service should be independently developed and deployed. No
coordination should be needed with other service teams if no
breaking API changes have been made. Each service is effectively
it's own product with it's own codebase and lifecycle.

                     ----
                   /~@@~\,
 _______ . _\_\___/\ __ /\___|_|_ . _______
/ ____  |=|       \  <_+>  /       |=|  ____ \
~|    |\|=|======\\_____//======|=|/|    |~
 |_   |    \       |       |       /    |    |
  \==-|     \      |  mX4  |      /     |----|~~)
   |   |      |    |       |    |       |____/~/
   |   |       \____\____/____/       /    / /
   |   |         {----------}        /____/ /
   |___|        /~~~~~~~~~~~~\      |_/~|_|/
    \_/        [/~~~~~||~~~~~\]      /__|\
    | |          |    ||||    |     (/|[[\)
    [_]         |      |  |     |
                |_____|  |_____|
                (_____)  (_____)
                |     |  |     |
                |     |  |     |
                |/~~~\|  |/~~~\|
                /|___|\  /|___|\
               <_______><_______>

 A microservice architecture shifts around complexity. Instead of
a single complex system, you have a bunch of simple services with
complex interactions.
```

```
637:   Doc:
638:   https://rapidapi.com/hub
639:   http://text-processing.com/demo/sentiment/
640:   https://opensource.com/resources/what-are-microservices
641:   https://sourceforge.net/projects/alcinoe/
642:   http://fundementals.sourceforge.net/unicode.html
643:
       http://www.softwareschule.ch/examples/1142_list_collections_pydemo42.txt
644:
       http://www.softwareschule.ch/examples/1145_271_closures_study_op_sys_tut
645:
646: https://github.com/LibreTranslate/LibreTranslate#mirrors
647: -----_____
```



```
648:
649:
650:
651:
652:
653:
654:
655:
656:
657:
658:
659:
660:
661:
662:
663:
664:
665: namespace RegExApplication
666:     {
667:         class Program
668:         {
669:             private static void showMatch(string text, string expr)
670:             {
671:                 int arex = 46;
672:                 Console.WriteLine("The Expression: " + expr + '
       '+arex);
673:                 MatchCollection mc = Regex.Matches(text, expr);
674:                 foreach (Match m in mc)
675:                 {
676:                     Console.WriteLine(m);
677:                 }
678:             }
679:             static void Main(string[] args)
680:             {
681:                 string str = "A Thousand Splendid Suns";
682:                 Console.WriteLine("Matching words that start with
       'S': ");
683:                 showMatch(str, @"\bS\S*");
684:                 FileIOApplication.FileProgram.Mainfiler();
685:                 BinaryFileApplication.Program.Mainbinary();
686:                 WindowsFileApplication.Program.Mainwin();
687:                //
       DelegateAppl.BoilerEventAppl.RecordBoilerInfo.Mainboiler();
```

```
688:                 //DelegateAppl.TestDelegate.MainDelegate(["Arg 1",
    "Arg 2", "Arg 3"]);
689:                 //  DelegateAppl.TestDelegate.MainDelegate("Arg
    3");
690:
691:                 //System.Windows.Input.ICommand.Equals.
692:
693:                 Console.ReadKey();
694:                 foreach (var arg in args)
695:                 {
696:                     Console.WriteLine(arg);
697:                 }
698:                 /* for(int i = 0; i < args.length; i++) {
699:                     Console.WriteLine(args[i]);
700:                     }
701:                 Console.ReadKey();*/
702:                 //205
703:             }
704:         }
705:     }
706:
707: def generate_power_func(n):
708:     print "id(n): %X" % id(n)
709:     def nth_power(x):
710:         return x**n
711:     print "id(nth_power): %X" % id(nth_power)
712:     return nth_power
713:
714: >>> raised_to_4 = generate_power_func(4)
715: id(n): CCF7DC
716: id(nth_power): C46630
717: >>> repr(raised_to_4)
718: '<function nth_power at 0x00C46630>'
719:
720: def generate_power_func(n):
721:     print "id(n): %X" % id(n)
722:     def nth_power(x):
723:         return x**n
724:     print "id(nth_power): %X" % id(nth_power)
725:     return nth_power
726:
727: >>> raised_to_4 = generate_power_func(4)
728: id(n): CCF7DC
729: id(nth_power): C46630
730: >>> repr(raised_to_4)
731: '<function nth_power at 0x00C46630>'
732:
733: maXbox4
734: Total of Function Calls: 35771
735: SHA1: 4.7.6.10 30bace36b037686509bbbee256e22daa52b3df70
736: CRC32: 500F69DD: 31.8 MB (33,400,088 bytes)
737: ZIP maxbox4.zip SHA1: 9DA15BFD72471108FCF746669C6AFD96E9A0E54C
738:
```