```
#!/bin/bash
# Encoding: UTF-8 (Unicode)
# :set syntax=sh
#
#===========================================================================
# ©MMXX- J. S. Gilstrap – All Rights Reserved.
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; version 3 of the License.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <http://www.gnu.org/licenses/>.
#
# Any knowledge gained by viewing and/or using this software may NOT be used
# to file for any 'submarine' patents on applicable algorithmns and/or ideas
# or any other method that would undermine the free use and/or distribution
# thereof. By viewing and/or using this software you agree to these terms.
# If you cannot agree to this then do NOT view and/or use this software.
#===========================================================================

# OZ Shoe Wizard

# Display help.
if [ "$1" = --help ]
then
      echo
      echo "⇐[1m         Oz Shoe Wizard™⇐[0m"
      echo      " Usage: oz-shoe-wizard [ output filename ]"
      echo
      exit 1
fi

# Text Editors
EDITORS="vi vim gvim emacs pico nano gedit kedit kate"

# ANSI Color Formatting.
# ESC character is encoded as '⇐'. Use find/replace to insert the real character.
# In vi use Ctrl-V Ctrl-[ –> :1,$s+⇐+^[+g

Mx="⇐[30;42mM⇐[47me⇐[41mx⇐[0m" ; UK="⇐[1;38;5;129mUK⇐[0m" ; US="⇐[1;34mU⇐[31mS⇐[0m"
usM="⇐[1;34mu⇐[31ms⇐[32mM⇐[0m" ; usW="⇐[1;34mu⇐[31ms⇐[38;5;124mW⇐[0m"
BR="⇐[1;48;5;40;38;5;21m B⇐[33mR ⇐[0m" ; ITEU="⇐[1;31mI⇐[32mT⇐[0m|⇐[1;33;44mEU⇐[0m" ;
FR="⇐[37;44mF⇐[47m ⇐[41mR⇐[0m"
# Uncomment for no color.
#Mx="⇐[1mMex⇐[0m" ; UK="⇐[1mUK⇐[0m" ; usM="⇐[1musM⇐[0m" ; usW="⇐[1musW⇐[0m" ; BR="
⇐[1mBR⇐[0m " ; ITEU="⇐[1mIT|EU⇐[0m" ; FR="⇐[1mFR⇐[0m "

# Display message.
cat <<End

      ⇐[1;33;42m Oz Shoe Wizard ⇐[0m

Set Terminal Geometry to 84×40 minimum.

Paris Point to Barleycorn Last conversion accuracy is
within ±½mm from sizes usW 5 to usM 11 (UK 2½ to 10)
[Mx|UK|usM|usW] = ⅘ × IT|EU — [26|25½|24½|23]
```

```
An online version can be found here:
https://www.amstzone.org/Home/shoe-size-select.html
An abridged version is here:
https://www.amstzone.org/Home/oz-shoe-wizard.html

Width & Girth calculations are tuned to data obtained from
DOROTHY Mass Foot Measurement Campaign of Europe in 2010.
https://www.3dbodyscanning.org/cap/papers/2010/10338_31jurca.pdf

⇐[1mMondoPoint Length, Width & Girth values are barefoot measurements.
Depending on sock thickness these can increase.⇐[0m

End

# Prompt for output filename.
unset fname
while :
do
      printf "\rOutput Filename <${1}> :  \b"
      read fname
      [ -n "$fname" ] && OutFile="$fname" || OutFile="$1"
      [ -n "$OutFile" ] && break
      printf "⇐[1;31m Enter Filename <> :  \b⇐[0m"
      sleep 1
done

# Open file handle for temporary file.
TMP=/tmp/oz-shoe-wizzard_$$
if exec 3> $TMP
then
      printf "\n$TMP temporary file created.\n\n"
else
      printf "\n⇐[1;5;31mTemp file Can NOT be created. Quitting!⇐[0m\n"
      sleep 1
      exit 1
fi

while :
do

# Column data processor.
Parse()
      {
      local cnt=0 value label

      # Gather values
      for value in $VALUES
      do
            label=${COLUMNS[$cnt]}
            case "$label" in
                      UK:)  width="$value" ;;
                      mm:) length="$value" ;;
                      in.)     in="$value" ;;
                  Last.mm)   last="$value" ;;
            esac
            let cnt++
      done

      cnt=0
      echo >&3
      # Print values
      for value in $VALUES
      do
            label=${COLUMNS[$cnt]}
            [ -z "$label" ] && break
            printf "% 15b" "${label/_/ }" >&3

            case "$label" in
                  'IT|EU:')
                        case "$System" in
                                    Barleycorn) value="$value  = 1¼ × [ ${length}mm ÷ 8 +
½ ] (Foot Length)" ;;
                              'MondoPoint 5mm') value="$value  = 0.16 × ${length}mm (Foot
Length)" ;;
                                             *) value="$value  = 4  × $in\" (Foot Length)"
;;
                        esac
                  ;;
                         CM:) value="$value  = 2⅔ × $in\" (Foot Length)" ;;
                  of_Length:) value="${value}%" ;;
                      Label:) value="${value/_/ }" ;;
            esac

            echo " $value" >&3
            let cnt++
      done
      }

# Yes/No Dialog.
qYN()
      {
      local yn
      printf "$1 [y|n] : "
      read yn ; [ "$yn" = y ] && return 0 || return 1
      }

# Check Value.
ckVal()
      {
      if [ -z "$VALUES" ]
      then
            echo "      ⇐[1;5;31mEnter an Available Index Number.⇐[0m"

      elif [ "$VALUES" = Quit ]
      then
            eXit
      else
            break
      fi
      }

# Edit Text File.
editFile()
      {
      while :
      do
            printf "\nSelect Editor\n"

            select Ed in $EDITORS
            do
                  [ -n "$Ed" ] && break
            done

            which $Ed > /dev/null 2>&1 && break
            EDITORS=${EDITORS/$Ed/}
            printf "\n  ⇐[1;31mText editor '$Ed' is not available on this system.⇐[0m\n\n"
            sleep 1
      done

      $Ed "$1"
      echo
      }

# Create Output and Exit Clean.
eXit()
      {
      local pwd
      echo >&3 ; exec 3<&-
      [ "$1" != save ] && { rm $TMP ; exit 1 ; }

      clear ; cat $TMP
      echo

      if qYN "Create Text Output?"
      then
            while :
            do
                  unset pwd
                  printf "Save  ${OutFile}.txt  in  <${PWD}> ? : "
                  read  pwd

                  if [ -n "$pwd" ]
                  then
                        if [ -d "$pwd" ] && touch ${pwd}/isWriteable 2> /dev/null
                        then
                              rm ${pwd}/isWriteable
                              CWD="$pwd"
                              break
                        else
                              echo "⇐[1;5;31m   Directory not available.⇐[0m"
                        fi
                  else
                        CWD="$PWD"
                        break
                  fi
            done

            SAVE="${CWD}/${OutFile}.txt"

            if mv $TMP "$SAVE" > /dev/null 2>&1
            then
                  echo "⇐[1m      ${OutFile}.txt created.⇐[0m"
                  qYN "Edit  ${OutFile}.txt ?" && \
                  { export CWD ; editFile "$SAVE" ; }

                  if qYN "Print ${OutFile}.txt to Default Printer?"
                  then
                        if lpr "$SAVE"
                        then
                              echo "Sent to Print Queue."
                        else
                              echo "⇐[1;5;31mFailed to Print.⇐[0m"
                              sleep 1
                              exit 1
                        fi
                  fi
            else
```

```
                  echo "⇐[1;5;31mFailed to Create Text File.⇐[0m" >&2
                  sleep 1
                  exit 1
            fi
      else
            rm $TMP > /dev/null 2>&1
      fi

      sleep 1
      exit
      }

# Add Title.
unset Ttl ; printf "Specify document title. <${OutFile//_/ }> : "
read  Ttl ; [ -n "$Ttl" ] && Title="$Ttl" || Title="${OutFile//_/ }"
printf "\n$Title\n" >&3 ; l=${#Title} ; l=$(($l+1)) ; tildeline=`printf "% ${l}b" "\n"` ;
echo ${tildeline// /\~} >&3

# Gender Selection.
printf "\n⇐[1mSelect Gender⇐[0m\n"
select Gen in Male Female Unisex Quit
do
      if [ -z "$Gen" ]
      then
            echo "      ⇐[1;5;31mEnter a Correct Index Number.⇐[0m"

      elif [ "$Gen" = Quit ]
      then
            eXit
      else
            break
      fi
done

echo   "        Gender: ${Gen/_/ }" >&3
[ "$Gen" = Unisex ] && printf "\nUnsex profile is an average of the male & female
profiles.\n"

# System Selection.
printf "\n⇐[1mSelcet Sizing System.⇐[0m\n"
select System in Barleycorn "Paris Point" "MondoPoint 5mm" "½cm Last Inc." Quit
do
      if [ -z "$System" ]
      then
            echo "      ⇐[1;5;31mEnter a Correct Index Number.⇐[0m"

      elif [ "$System" = Quit ]
      then
            eXit
      else
            break
      fi
done
printf "\n Sizing System: ${System}\n" >&3

##################
# Size Selection #
##################
clear
# Paris Point 3⅓mm Last Increment, Foot increment ⅛".
if [ "$System" = "Paris Point" ]
then
      COLUMNS=(BR: IT\|EU: FR: Mx: UK: usM: usW: in. mm:)
      printf "\n          Size, 5%% Toe Room" >&3
      printf "\n⇐[1mParis Point Sizes, +5%% Toe Room⇐[0m\n"
      echo "    Select Size                                         Foot Length"
      echo "    $BR   $ITEU   $FR    $Mx     $UK      $usM    $usW     in.     mm"

select VALUES in \
" 31     33     34      0⅖     0.9     1.9     3⅖      8¼    209.6" \
" 31½    33½    34½     0⅘     1.3     2.3     3⅘      8⅜    212.7" \
" 32     34     35      1⅕     1.7     2.7     4⅕      8½    215.9" \
" 32½    34½    35½     1⅗     2.1     3.1     4⅗      8⅝    219.1" \
" 33     35     36      2      2½      3½      5       8¾    222.3" \
" 33½    35½    36½     2⅖     2.9     3.9     5⅖      8⅞    225.4" \
" 34     36     37      2⅘     3.3     4.3     5⅘      9     228.6" \
" 34½    36½    37½     3⅕     3.7     4.7     6⅕      9⅛    231.8" \
" 35     37     38      3⅗     4.1     5.1     6⅗      9¼    235.0" \
" 35½    37½    38½     4      4½      5½      7       9⅜    238.1" \
" 36     38     39      4⅖     4.9     5.9     7⅖      9½    241.3" \
" 36½    38½    39½     4⅘     5.3     6.3     7⅘      9⅝    244.5" \
" 37     39     40      5⅕     5.7     6.7     8⅕      9¾    247.7" \
" 37½    39½    40½     5⅗     6.1     7.1     8⅗      9⅞    250.8" \
" 38     40     41      6      6½      7½      9      10     254.0" \
" 38½    40½    41½     6⅖     6.9     7.9     9⅖     10⅛    257.2" \
" 39     41     42      6⅘     7.3     8.3     9⅘     10¼    260.4" \
" 39½    41½    42½     7⅕     7.7     8.7    10⅕    10⅜    263.5" \
" 40     42     43      7⅗     8.1     9.1    10⅗    10½    266.7" \
" 40½    42½    43½     8      8½      9½     11      10⅝    269.9" \
" 41     43     44      8⅖     8.9     9.9    11⅖    10¾    273.1" \
" 41½    43½    44½     8⅘     9.3    10.3    11⅘     10⅞    276.2" \
" 42     44     45      9⅕     9.7    10.7    12⅕     11     279.4" \
" 42½    44½    45½     9⅗    10.1    11.1    12⅗     11⅛    282.6" \
" 43     45     46     10     10½     11½     13      11¼    285.8" \
" 43½    45½    46½    10⅖    10.9    11.9    13⅖     11⅜    288.9" \
" 44     46     47     10⅘    11.3    12.3    13⅘     11½    292.1" \
" 44½    46½    47½    11⅕    11.7    12.7    14⅕     11⅝    295.3" \
" 45     47     48     11⅗    12.1    13.1    14⅗     11¾    298.5" \
" 45½    47½    48½    12     12½     13½     15      11⅞    301.6" \
" 46     48     49     12⅖    12.9    13.9    15⅖     12     304.8" \
" 46½    48½    49½    12⅘    13.3    14.3    15⅘     12⅛    308.0" \
" 47     49     50     13⅕    13.7    14.7    16⅕     12¼    311.2" \
" 47½    49½    50½    13⅗    14.1    15.1    16⅗     12⅜    314.3" \
" 48     50     51     14     14½     15½     17      12½    317.5" Quit
do
      ckVal
done
      Parse

# Barleycorn ~⅙" Last Increment, Foot Increment 4mm.
elif [ "$System" = Barleycorn ]
then
      COLUMNS=(Mx: UK: usM: usW: BR: IT\|EU: FR: mm: in. Last.mm:)
      printf "\n          Size, 5⅚%% Toe Room" >&3
      printf "\n⇐[1mBarleycorn Sizes, +5⅚%% Toe Room⇐[0m\n"
      echo "    Select Size                                          Foot Length"
      echo "     $Mx    $UK     $usM    $usW   $BR    $ITEU    $FR      mm     in.
Last.mm"

select VALUES in \
"  –½     0      1      2½    29⅞     31⅞     32⅞     200     7.87     211.2" \
"  0       ½      1½     3     30½     32½     33½     204     8.03     215.5" \
"   ½     1      2      3½    31⅛     33⅛     34⅛     208     8.19     219.7" \
"  1      1½     2½     4     31¾     33¾     34¾     212     8.34     224  " \
"  1½     2      3      4½    32⅜     34⅜     35⅜     216     8.50     228.3" \
"  2      2½     3½     5     33      35      36      220     8.66     232.5" \
"  2½     3      4      5½    33⅝     35⅝     36⅝     224     8.82     236.8" \
"  3      3½     4½     6     34¼     36¼     37¼     228     8.97     241.1" \
"  3½     4      5      6½    34⅞     36⅞     37⅞     232     9.13     245⅓ " \
"  4      4½     5½     7     35½     37½     38½     236     9.29     249.6" \
"  4½     5      6      7½    36⅛     38⅛     39⅛     240     9.45     253.9" \
"  5      5½     6½     8     36¾     38¾     39¾     244     9.61     258.1" \
"  5½     6      7      8½    37⅜     39⅜     40⅜     248     9.76     262.4" \
"  6      6½     7½     9     38      40      41      252     9.92     266⅔ " \
"  6½     7      8      9½    38⅝     40⅝     41⅝     256    10.08     270.9" \
"  7      7½     8½    10     39¼     41¼     42¼     260    10.23     275.2" \
"  7½     8      9     10½    39⅞     41⅞     42⅞     264    10.39     279.5" \
"  8      8½     9½    11     40½     42½     43½     268    10.55     283.7" \
"  8½     9     10     11½    41⅛     43⅛     44⅛     272    10.71     288  " \
"  9      9½    10½    12     41¾     43¾     44¾     276    10.86     292.3" \
"  9½    10     11     12½    42⅜     44⅜     45⅜     280    11.02     296.5" \
" 10     10½    11½    13     43      45      46      284    11.18     300.8" \
" 10½    11     12     13½    43⅝     45⅝     46⅝     288    11.34     305.1" \
" 11     11½    12½    14     44¼     46¼     47¼     292    11.49     309⅓ " \
" 11½    12     13     14½    44⅞     46⅞     47⅞     296    11.65     313.6" \
" 12     12½    13½    15     45½     47½     48½     300    11.81     317.9" \
" 12½    13     14     15½    46⅛     48⅛     49⅛     304    11.97     322.1" \
" 13     13½    14½    16     46¾     48¾     49¾     308    12.12     326.4" \
" 13½    14     15     16½    47⅜     49⅜     50⅜     312    12.28     330.7" \
" 14     14½    15½    17     48      50      51      316    12.44     334.9" \
" 14½    15     16     17½    48⅝     50⅝     51⅝     320    12.60     339.2" Quit
do
      ckVal
done
      Parse

# MondoPoint 5mm Foot Increment, 5⅓mm Last Increment.
elif [ "$System" = "MondoPoint 5mm" ]
then
      COLUMNS=(mm: BR: IT\|EU: FR: Mx: UK: usM: usW: in. Last.mm:)
      printf "\n          Size, 6⅔%% Toe Room" >&3
      printf "\n⇐[1mMondoPoint Sizes, +6⅔%% Toe Room⇐[0m\n"
      echo "    Select Size                                                Foot Length"
      echo "      ⇐[1mMM⇐[0m    $BR   $ITEU    $FR     $Mx      $UK      $usM     $usW
in.   Last.mm"

select VALUES in \
" 205    30.8    32.8    33.8     0.2     0.7     1.7     3.2     8.07    218⅔" \
" 210    31.6    33.6    34.6     0.9     1.4     2.4     3.9     8.27    224 " \
" 215    32.4    34.4    35.4     1.5     2.0     3.0     4.5     8.46    229⅓" \
" 220    33.2    35.2    36.2     2.2     2.7     3.7     5.2     8.66    234⅔" \
" 225    34.0    36.0    37.0     2.8     3.3     4.3     5.8     8.86    240 " \
" 230    34.8    36.8    37.8     3.4     3.9     4.9     6.4     9.06    245⅓" \
" 235    35.6    37.6    38.6     4.1     4.6     5.6     7.1     9.25    250⅔" \
" 240    36.4    38.4    39.4     4.7     5.2     6.2     7.7     9.45    256 " \
" 245    37.2    39.2    40.2     5.4     5.9     6.9     8.4     9.65    261⅓" \
" 250    38.0    40.0    41.0     6.0     6.5     7.5     9.0     9.84    266⅔" \
" 255    38.8    40.8    41.8     6.6     7.1     8.1     9.6    10.04    272 " \
" 260    39.6    41.6    42.6     7.3     7.8     8.8    10.3    10.24    277⅓" \
" 265    40.4    42.4    43.4     7.9     8.4     9.4    10.9    10.43    282⅔" \
" 270    41.2    43.2    44.2     8.6     9.1    10.1    11.6    10.63    288 " \
" 275    42.0    44.0    45.0     9.2     9.7    10.7    12.2    10.83    293⅓" \
" 280    42.8    44.8    45.8     9.8    10.3    11.3    12.8    11.02    298⅔" \
" 285    43.6    45.6    46.6    10.5    11.0    12.0    13.5    11.22    304 " \
" 290    44.4    46.4    47.4    11.1    11.6    12.6    14.1    11.42    309⅓" \
" 295    45.2    47.2    48.2    11.8    12.3    13.3    14.8    11.61    314⅔" \
" 300    46.0    48.0    49.0    12.4    12.9    13.9    15.4    11.81    320 " \
" 305    46.8    48.8    49.8    13.0    13.5    14.5    16.0    12.01    325⅓" \
" 310    47.6    49.6    50.6    13.7    14.2    15.2    16.7    12.20    330⅔" \
" 315    48.4    50.4    51.4    14.3    14.8    15.8    17.3    12.40    336 " \
" 320    49.2    51.2    52.2    15.0    15.5    16.5    18.0    12.60    341⅓" Quit
do
      ckVal
done
      Parse

# ½ Centimeter Last Increment, 3/16" Foot Increment.
elif [ "$System" = "½cm Last Inc." ]
then
      COLUMNS=(CM: BR: IT\|EU: FR: Mx: UK: usM: usW: in. mm:)
      printf "\n          Size, 5%% Toe Room" >&3
      printf "\n⇐[1m½ Centimeter Sizes, +5%% Toe Room⇐[0m\n"
      echo "    Select Size                                                     Foot
Length"
      echo "     ⇐[1mCM⇐[0m    $BR   $ITEU   $FR     $Mx     $UK      $usM     $usW
in.        mm"

select VALUES in \
" 22     31     33     34      0.4     0.9     1.9     3.4      8¼        209.6" \
" 22½    31¾    33¾    34¾     1.0     1.5     2.5     4.0      8⁷/₁₆     214.3" \
" 23     32½    34½    35½     1.6     2.1     3.1     4.6      8⅝        219.1" \
" 23½    33¼    35¼    36¼     2.2     2.7     3.7     5.2      8¹³/₁₆    223.8" \
" 24     34     36     37      2.8     3.3     4.3     5.8      9         228.6" \
" 24½    34¾    36¾    37¾     3.4     3.9     4.9     6.4      9³/₁₆     233.4" \
" 25     35½    37½    38½     4.0     4.5     5.5     7.0      9⅜        238.1" \
" 25½    36¼    38¼    39¼     4.6     5.1     6.1     7.6      9⁹/₁₆     242.9" \
" 26     37     39     40      5.2     5.7     6.7     8.2      9¾        247.7" \
" 26½    37¾    39¾    40¾     5.8     6.3     7.3     8.8      9¹⁵/₁₆    252.4" \
" 27     38½    40½    41½     6.4     6.9     7.9     9.4     10⅛        257.2" \
" 27½    39¼    41¼    42¼     7.0     7.5     8.5    10.0     10⁵/₁₆     261.9" \
" 28     40     42     43      7.6     8.1     9.1    10.6     10½        266.7" \
" 28½    40¾    42¾    43¾     8.2     8.7     9.7    11.2     10¹¹/₁₆    271.5" \
" 29     41½    43½    44½     8.8     9.3    10.3    11.8     10⅞        276.2" \
" 29½    42¼    44¼    45¼     9.4     9.9    10.9    12.4     11¹/₁₆     281.0" \
" 30     43     45     46     10.0    10.5    11.5    13.0     11¼        285.8" \
" 30½    43¾    45¾    46¾    10.6    11.1    12.1    13.6     11⁷/₁₆     290.5" \
" 31     44½    46½    47½    11.2    11.7    12.7    14.2     11⅝        295.3" \
" 31½    45¼    47¼    48¼    11.8    12.3    13.3    14.8     11¹³/₁₆    300.0" \
" 32     46     48     49     12.4    12.9    13.9    15.4     12         304.8" \
" 32½    46¾    48¾    49¾    13.0    13.5    14.5    16.0     12³/₁₆     309.6" \
" 33     47½    49½    50½    13.6    14.1    15.1    16.6     12⅜        314.3" \
" 33½    48¼    50¼    51¼    14.2    14.7    15.7    17.2     12⁹/₁₆     319.1" Quit
do
      ckVal
done
      Parse
else
      eXit
fi

###################
# Widths & Girths #
###################

# Width & Girth Parameters.
# WIDTH|GIRTH % of Length: .#### Tangent & _BIAS: ##.## mm.
case $Gen in
      Male)  # A    B    C    D    E    F    G    H    I    J    K
        WIDTH=(2328 2391 2454 2520 2587 2656 2727 2800 2874 2951 3030)
        WBIAS=(2773 2847 2923 3001 3081 3163 3248 3334 3423 3514 3608)  # Updated
        GIRTH=(6044 6159 6276 6395 6516 6640 6766 6894 7025 7158 7294)  # Parameters
        GBIAS=(7199 7335 7474 7616 7761 8086 8058 8211 8367 8525 8687)
```

```
       #HWDTH=()
       #HBIAS=()
            # (2297 2364 2433 2504 2577 2652 2730 2809 2891 2976 3063)
            # (2682 2761 2841 2924 3009 3097 3188 3281 3376 3475 3576)  # Original
            # (5815 5965 6119 6277 6439 6605 6775 6950 7129 7313 7502)  # Parameters
            # (7035 7216 7402 7593 7789 7990 8196 8407 8624 8847 9075)
      ;;
      # The Unisex array has been updated and is now a middle average of the Male & Female
Arrays.
      Unisex)
        WIDTH=(2948 3029 3112 3197 3285 3375 3468 3563 3660 3761 3864)
        WBIAS=(1059 1088 1117 1148 1180 1212 1245 1279 1314 1350 1388)  # Updated
        GIRTH=(7606 7761 7920 8081 8246 8414 8585 8760 8939 9121 9307)  # Parameters
        GBIAS=(2731 2787 2844 2902 2961 3021 3083 3146 3210 3275 3342)
       #HWDTH=()
       #HBIAS=()
            # (2413 2484 2556 2631 2707 2786 2868 2951 3038 3126 3217)
            # (2310 2378 2447 2519 2592 2668 2746 2826 2908 2993 3081)  # Original
            # (5672 5818 5968 6122 6280 6442 6608 6778 6953 7132 7316)  # Parameters
            # (7112 7295 7483 7676 7874 8077 8286 8499 8718 8943 9174)
      ;;
      Female)
        WIDTH=(2537 2608 2682 2757 2835 2915 2997 3082 3168 3258 3349)
        WBIAS=(1966 2021 2078 2137 2197 2259 2322 2388 2455 2524 2595)  # Updated
        GIRTH=(6494 6637 6784 6934 7087 7244 7404 7568 7735 7906 8081)  # Parameters
        GBIAS=(5032 5143 5257 5373 5492 5613 5737 5864 5994 6126 6262)
       #HWDTH=()
       #HBIAS=()
            # (2529 2603 2679 2757 2838 2921 3006 3094 3184 3277 3372)
            # (1938 1995 2053 2113 2175 2238 2304 2371 2440 2511 2585)  # Original
            # (5528 5670 5817 5967 6121 6279 6440 6607 6777 6952 7131)  # Parameters
            # (7188 7374 7564 7759 7959 8164 8375 8591 8813 9040 9273)
      ;;
esac

# Calculate Width & Girth.
[ ${#length} -gt 3 ] && l=${length/./} || l="${length}0"
Q=0
for L in {A..K}
do
       w=${WIDTH[$Q]}
      ow=${WBIAS[$Q]}
       g=${GIRTH[$Q]}
      og=${GBIAS[$Q]}
      # h=${HWDTH[$Q]}
      #oh=${HBIAS[$Q]}

      w=$((  $w * $l + ${ow}000 ))
      p=$(( $w / $l + 5 ))
      g=$((  $g * $l + ${og}000 ))
      r=$(( 1000 * $g / $w + 5 ))
      w=$(( ($w + 5000) / 10000 ))
      g=$(( ($g + 5000) / 10000 ))
      p=${p:0:2}.${p:2:1}
      #h=$((  $h * $l + ${oh}000 ))
      #h=$(( ($h + 5000) / 10000 ))

      eval  $L=\$p
      eval w$L=\$w
      eval g$L=\$g
      eval r$L=\$r
      #eval h$L=\$h
      ((Q++))
```

```
done

procWidth()
      {
      if [ -z "$VALUES" ]
      then
            echo "      ⇐[1;5;31mEnter an Available Index Number.⇐[0m"

      elif [ "${VALUES:1:4}" = Quit ]
      then
            eXit save
      else
            break
      fi
      }

N=$C ; M=$E ; W=$G ; X=$I
# Width Selection.
clear
COLUMNS=(of_Length: US: UK: Label: Alt.)
printf "\n⇐[1mShoe Widths⇐[0m\n"
echo "Select Width (Percent of Foot Length)"
echo "    Percent      $US      $UK        Label       Alt."

if [ $Gen = Male ]
then
      select VALUES in \
      " $A        AA       B         Slim        II                           " \
      " $B         A       C       Narrow       III                           " \
      " $C         B       D       NarMed        IV                           " \
      " $D         C       E       Medium         V                           " \
      " $E         D       F       WidMed        VI                           " \
      " $F         E       G         Wide       VII                           " \
      " $G        EE       H      2X_Wide      VIII                           " \
      " $H       EEE       I      3X_Wide        IX                           " \
      " $I        4E       J      4X_Wide         X                           " \
      " $J        5E       K      5X_Wide        XI                           " \
      " $K        6E       L      6X_Wide       XII                           " \
      " $N       Nar       N       Narrow       III                           " \
      " $M       Med       M       Medium         V                           " \
      " $W       Wid       W         Wide       VII                           " \
      " $X       2XW       X      2X_Wide        IX                           " \
      " Quit                                                                  "
      do
            procWidth
      done
else
      select VALUES in \
      " $A       AAA       A       X_Slim         I                           " \
      " $B        AA       B         Slim        II                           " \
      " $C         A       C       Narrow       III                           " \
      " $D         B       D       NarMed        IV                           " \
      " $E         C       E       Medium         V                           " \
      " $F         D       F       WidMed        VI                           " \
      " $G         E       G         Wide       VII                           " \
      " $H        EE       H      2X_Wide      VIII                           " \
      " $I       EEE       I      2X_Wide        IX                           " \
      " $J        4E       J      4X_Wide         X                           " \
      " $K        5E       K      5X_Wide        XI                           " \
      " $N       Nar       N       Narrow       III                           " \
      " $M       Med       M       Medium         V                           " \
      " $W       Wid       W         Wide       VII                           " \
      " $X       2XW       X      2X_Wide        IX                           " \
      " Quit                                                            "
      do
            procWidth
      done
fi

printf "\n         Width" >&3

Parse

if [ $Gen =  Male ]
then
  case $width in
      B) width=A ;; C) width=B ;; D) width=C ;; E) width=D ;; F) width=E ;; G) width=F ;;
      H) width=G ;; I) width=H ;; K) width=I ;; K) width=J ;; L) width=K ;;
  esac
fi

# Set Width & Girth.
case $width in
      A) w=$wA ; g=$gA ; r=$rA ; h=$hA ;; B) w=$wB ; g=$gB ; r=$rB ; h=$hB ;; C) w=$wC ;
g=$gC ; r=$rC ; h=$hC ;;
      D) w=$wD ; g=$gD ; r=$rD ; h=$hD ;; E) w=$wE ; g=$gE ; r=$rE ; h=$hE ;; F) w=$wF ;
g=$gF ; r=$rF ; h=$hF ;;
      G) w=$wG ; g=$gG ; r=$rG ; h=$hG ;; H) w=$wH ; g=$gH ; r=$rH ; h=$hH ;; I) w=$wI ;
g=$gI ; r=$rI ; h=$hI ;;
      J) w=$wJ ; g=$gJ ; r=$rJ ; h=$hJ ;; K) w=$wK ; g=$gK ; r=$rK ; h=$hK ;; N) w=$wC ;
g=$gC ; r=$rC ; h=$hC ;;
      M) w=$wE ; g=$gE ; r=$rE ; h=$hE ;; W) w=$wG ; g=$gG ; r=$rG ; h=$hG ;; X) w=$wI ;
g=$gI ; r=$rI ; h=$hI ;;
esac

# Print MondoPoint Length, Width, and Girth Values.
c=$(( ${#w} - 1 ))
printf "\n    MondoPoint (Weight Bearing)\n" >&3
echo     "        Length: ${l:0:3}.${l:3:1} mm" >&3
printf   "         Width: " >&3 ; printf "%9b" "${w:0:$c}.${w:$c:1} mm\n" >&3
echo     "         Girth: ${g:0:3}.${g:3:1} mm" >&3
echo     "     G÷W Ratio:   ${r:0:1}.${r:1:2}" >&3
#echo     "    Heel Width:  ${h:0:2}.${h:2:1} mm" >&3

if qYN "All OK?"
then
      break
else
      qYN "Redo?" || eXit
fi

done

# Add Notes.
if qYN "Add any Notes?"
then
      printf "\nNOTE: To End use Ctrl-D on a New Line.\n\n"
      printf "\nNotes:\n" >&3
      cat >&3
fi

eXit save
exit 1
```