**Function** LastWord(txt **As** **String**, Optional delim **As** **String** = " ", Optional n **As** **Integer** = 1) **As** **String**

LastWord = **Mid**(txt, InStrRev(txt, delim) + 1, 50)

**End** **Function**

Programmas kods Step1, pēc pogas noklikšķināšanas [SOLIS 1] tiek parādīts mājiens

**Sub** Step1()

**Dim** msg **As** **Integer**

msg = **MsgBox**("Ievietojiet datus lapā vai ielādējiet tos, atverot cilni Dati – Ārējo datu iegūšana. Pēc datu ielādes atgriezieties uz lapu rezultāts un noklikšķiniet uz [SOLIS 2] ", 0," Ielādēt datus Excel darbgrāmatā")

Sheets.Add.Name = "dati"

**End** **Sub**

Programmas kods Step2, datu apstrāde, sākas pēc pogas [SOLIS 2] nospiešanas

**Sub** Step2()

Sheets("dati").Select

*' kolonnas ar uzvārdiem kopēšana uz citu lapu*

Cells.Find(What:="Uzv?rds", After:=ActiveCell, LookIn:=xlFormulas, \_

LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:=xlNext, \_

MatchCase:=**False**, SearchFormat:=**False**).Activate

Range(ActiveCell, Cells(Rows.Count, ActiveCell.Column)).Copy

Sheets.Add.Name = "saraksts"

ActiveSheet.Paste

' lieku atstarpju noņemšana (dubultajos uzvārdos)

Range(ActiveSheet.Columns(1).Find("\*", , xlFormulas, xlWhole), Cells(Rows.Count, 1).**End**(xlUp)).**Select**

**Dim** rCell **As** Range

**For** **Each** rCell In Selection

rCell = Replace(rCell, " -", "-")

rCell = Replace(rCell, "- ", "-")

**Next**

' uzvārdu bez vārda kopēšana blakus šūnā

**For** i1 = 1 **To** Cells(Rows.Count, 1).**End**(xlUp).Row

Cells(i1, 2) = LastWord(Cells(i1, 1))

**Next**

' dubultuzvārdu atdalīšana ar atstarpi un katra vārda pirmo rakstzīmi pārvēršana par Lielo Reģistru

Range(ActiveSheet.Columns(2).Find("\*", , xlFormulas, xlWhole), Cells(Rows.Count, 2).**End**(xlUp)).**Select**

**Dim** fCell **As** Range

**For** **Each** fCell In Selection

fCell = Replace(fCell, "-", " ")

fCell = **StrConv**(fCell, vbProperCase)

**Next**

' otrā uzvārda izguve jaunā šūnā

**Dim** r **As** Range

**Set** r = Selection

**Dim** arr **As** **Variant**

**If** r.Cells.Count = 1 **Then**

**ReDim** arr(1 **To** 1, 1 **To** 1)

arr(1, 1) = r.Cells(1).Value

**Else**

arr = r

**End** **If**

**Dim** brr **As** **Variant**

**ReDim** brr(1 **To** **UBound**(arr, 1))

**Dim** y **As** **Long**

**Dim** u **As** **Long**

**Dim** v **As** **Variant**

**Dim** crr **As** **Variant**

**For** y = 1 **To** **UBound**(arr, 1)

**If** arr(y, 1) <> "" **Then**

crr = Split(arr(y, 1), " ")

**If** **Not** **IsEmpty**(crr) **Then**

**If** **UBound**(crr) > 0 **Then**

**ReDim** Preserve brr(1 **To** **UBound**(brr) + **UBound**(crr))

**End** **If**

**For** **Each** v In crr

u = u + 1

brr(u) = **Trim**(v)

**Next**

**End** **If**

**End** **If**

**Next**

**Set** r = r.Cells(1, 1).Resize(**UBound**(brr), 1)

r = Application.Transpose(brr)

**With** r.Parent.Sort

.SortFields.Clear

.SortFields.Add Key:=r, SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortTextAsNumbers

.SetRange r

.Header = xlNo

.MatchCase = **False**

.Orientation = xlTopToBottom

.SortMethod = xlPinYin

.Apply

**End** **With**

Range(ActiveCell, Cells(Rows.Count, ActiveCell.Column)).Copy

Sheets.Add.Name = "Uzvardi"

ActiveSheet.Paste

**Set** WS1 = ActiveSheet ' aktīva lapa

**Set** WS2 = ActiveWorkbook.Sheets("rezultats") ' - nepieciešamā lapa

WS2.Cells(1, 1).Value = WS2.Cells(1, 1).Value

WS2.Activate ' - nepieciešms palika aktiīvs

' rezultātu veidošana

**Dim** arrData, arrFIO, arrCounts, arrFIOCounts, Dict **As** **Object**, i **As** **Long**, n **As** **Long**, iRow **As** **Long**

**Dim** fourFirstLetters **As** **String**, allMatchFIO **As** **String**, Counter **As** **Long**, arrTemp

**Dim** Rng **As** Range, LastRow **As** **Long**, FirstLettersCount **As** **Long**, TOPNum **As** **Long**

TOPNum = 12 *' TOP10*

FirstLettersCount = 4 *' pirmo rakstzīmju skaits - kritērijs*

' ņemam datus no saraksts

**With** Worksheets("saraksts")

arrData = .Range("B2", .Cells(.Rows.Count, 2).**End**(xlUp)).Value2

**End** **With**

' skaitām katra uzvārda numuru vispārējā sarakstā

**Set** Dict = **CreateObject**("Scripting.Dictionary")

**For** i = 1 **To** **UBound**(arrData)

*' katrā uzvārda skaits*

*'Dict(arrData(i, 1)) = Dict(arrData(i, 1)) + 1*

*'skaits pēc kritērijām*

fourFirstLetters = **Left**(arrData(i, 1), FirstLettersCount)

Dict(fourFirstLetters) = Dict(fourFirstLetters) + 1

**Next** i

' veidojam masīvu Uzvārds-Daudzums

arrCounts = Dict.items

arrFIO = Dict.keys

**ReDim** arrFIOCounts(1 **To** **UBound**(arrFIO) + 1, 1 **To** 2)

**For** i = **LBound**(arrFIO) **To** **UBound**(arrFIO)

arrFIOCounts(i + 1, 1) = arrFIO(i)

arrFIOCounts(i + 1, 2) = arrCounts(i)

**Next** i

' kārtojam iegūto masīvu pēc atrastajiem daudzumiem dilstošā secībā, lai noteiktu visbiežāk izmantotos. Viņi atradīsies saraksta augšgalā

Application.ScreenUpdating = False

Workbooks.Add(1).Sheets(1).Range("A1").Resize(**UBound**(arrFIOCounts, 1), 2).Value2 = arrFIOCounts

**With** ActiveSheet

' šķirošana - dilstoši daudzumi

LastRow = .Cells(.Rows.Count, 1).**End**(xlUp).Row

**Set** Rng = .Range("A1").CurrentRegion

.Sort.SortFields.Clear

.Sort.SortFields.Add Key:=.Range("B1:B" & LastRow), SortOn:=xlSortOnValues, Order:=2, DataOption:=xlSortNormal

**With** .Sort

.SetRange Rng

.Header = xlNo

.MatchCase = False

.Orientation = xlTopToBottom

.SortMethod = xlPinYin

.Apply

**End** **With**

arrFIO = .Range("A1").Resize(**UBound**(arrFIOCounts), 2).Value

**End** **With**

ActiveWorkbook.Close (**False**)

' no biežāk lietotajiem atrodam 10 unikālus uzvārdus (pēc kritērija)

**Set** Dict = **CreateObject**("Scripting.Dictionary")

**For** i = 1 **To** **UBound**(arrFIO)

**If** Dict.Count = TOPNum **Then** **Exit** **For**

**If** **Not** Dict.exists(**Left**(arrFIO(i, 1), FirstLettersCount)) **Then**

Dict(**Left**(arrFIO(i, 1), FirstLettersCount)) = 0&

**End** **If**

**Next** i

arrTemp = Dict.keys

' veidojam 10 izplatītāko uzvārdu kopsavilkuma tabulu (pēc kritērija)

**With** ActiveSheet

**For** i = 0 **To** **UBound**(arrTemp)

iRow = iRow + 1

**Set** Dict = **CreateObject**("Scripting.Dictionary")

allMatchFIO = ""

Counter = 0

fourFirstLetters = arrTemp(i)

.Cells(iRow + 1, 1) = fourFirstLetters *' kritērijs*

' saskaitām, cik uzvārdu vispārējā sarakstā atbilst kritērijam

**For** n = 1 **To** **UBound**(arrData)

**If** **Left**(arrData(n, 1), FirstLettersCount) = fourFirstLetters **Then**

**If** **Not** Dict.exists(arrData(n, 1)) **Then**

Dict(arrData(n, 1)) = 0&

allMatchFIO = allMatchFIO & ", " & arrData(n, 1)

**End** **If**

Counter = Counter + 1

**End** **If**

**Next** n

.Cells(iRow + 1, 2) = Counter ' pierakstām atrasto kritērijam atbilstošu uzvārdu skaitu

.Cells(iRow + 1, 3) = Counter / **UBound**(arrData) ' aprēķinām % no kopējā uzvārdu skaita

.Cells(iRow + 1, 4) = **Mid**(allMatchFIO, 2) ' kritērijam atbilstošo unikālo uzvārdu saraksts

**Next** i

' kārtošana dilstošā secībā pēc kolonnas %

LastRow = .Cells(.Rows.Count, 1).**End**(xlUp).Row

**Set** Rng = .Range("A1").CurrentRegion

.Sort.SortFields.Clear

.Sort.SortFields.Add Key:=.Range("B1:B" & LastRow), SortOn:=xlSortOnValues, Order:=2, DataOption:=xlSortNormal

**With** .Sort

.SetRange Rng

.Header = xlYes

.MatchCase = **False**

.Orientation = xlTopToBottom

.SortMethod = xlPinYin

.Apply

**End** **With**

**End** **With**

Application.CutCopyMode = **False**

Application.ScreenUpdating = **True**

**End** **Sub**