```
Open "number_system_output.txt" For Output As #1
On Error GoTo err_handler

Cls
Screen 12

Dim Shared DIGITS%
DIGITS% = 32

Dim Shared ALL_CHRS$
ALL_CHRS$ = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ"

Dim Shared TRIM_LEADING_ZEROES%
TRIM_LEADING_ZEROES% = 1

Dim Shared ERR_DIGIT_UNREPRESENTABLE%
ERR_DIGIT_UNREPRESENTABLE% = 80

Dim Shared ERR_CHAR_UNREPRESENTABLE%
ERR_CHAR_UNREPRESENTABLE% = 81

Dim Shared ERR_OVERFLOW%
ERR_OVERFLOW% = 82

Dim Shared ERR_MSG$
ERR_MSG$ = ""

' START: Main driver code
lb
pl "..... NUMBER SYSTEM CONVERTER (Made by RC) ....."
Dim in_arr%(DIGITS%)
Dim res_arr%(DIGITS%)

main_start:
Call fill_arr(in_arr%(), 0, DIGITS%)
Call fill_arr(res_arr%(), 0, DIGITS%)
lb
in_num$ = in_str$(" -> Enter number to convert: ")
base_from% = in_int%(" -> From base: ")
If base_from% < 2 Or base_from% > 36 Then
  pl "ERROR: Base must be in range [2, 36]"
  GoTo main_start
End If

Call toArr(in_num$, base_from%, in_arr%())

base_to% = in_int%(" -> To base: ")
If base_to% < 2 Or base_to% > 36 Then
  pl "ERROR: Base must be in range [2, 36]"
  GoTo main_start
End If

step_count% = 1
result_str$ = ""
lb
If base_from% = base_to% Then
  pl "Please enter different bases!"
  GoTo main_start
Else
  dec&& = toDecimal&&(in_arr%(), base_from%)
  If base_from% <> 10 Then
    pl "STEP " + s$(step_count%) + ": base " + s$(base_from%) + " -> base " + s$(10) + " : " + s$(dec&&)
    step_count% = step_count% + 1
  End If

  If base_to% <> 10 Then
    Call convertDecimal(dec&&, base_to%, res_arr%())
    result_str$ = toString$(res_arr%(), "")
    pl "STEP " + s$(step_count%) + ": base " + s$(10) + " -> base " + s$(base_to%) + " : " + result_str$
    step_count% = step_count% + 1
  Else
    result_str$ = s$(dec&&)
  End If
End If

pl "RESULT: " + in_num$ + " (base " + s$(base_from%) + ") = " + result_str$ + " (base " + s$(base_to%) + ")"
GoTo main_start
' END: Driver code

err_handler:
If Err > 0 Then
  pl "ERROR: line " + s$(Erl) + "  Code: " + s$(Err) + "  Msg: " + ERR_MSG$
  ERR_MSG$ = ""

  Select Case Err
    Case ERR_DIGIT_UNREPRESENTABLE%
      pl "FATAL ERROR: Digit cannot be represented. Make sure it is >= 0 and < 36"
      GoTo main_start
    Case ERR_CHAR_UNREPRESENTABLE%
      pl "FATAL ERROR: char representation is invalid. Make sure it is < base and one of  " + ALL_CHRS$
      GoTo main_start
    Case ERR_OVERFLOW%
      pl "OVERLOW ERROR: cannot represent number completely. Ignoring..."
      Resume Next
    Case Else
      pl "FATAL ERROR: Unexpected Error. Quiting...."
      End
  End Select
End If

Sub fill_arr (arr%(), num%, length%)
  For i% = 1 To length%
    arr%(i%) = num%
  Next i%
End Sub

Sub copy_arr (src%(), dest%(), length%)
  For i% = 1 To length%
    dest%(i%) = src%(i%)
  Next i%
End Sub

Sub convertDecimal (num&&, b%, res%())
  v&& = Abs(num&&)
  For i% = 1 To DIGITS%
    If v&& > 0 Then
      res%(i%) = v&& Mod b%
      v&& = v&& \ b%
```

```
    Else
      res%(i%) = 0
    End If
  Next i%
End Sub

Function toDecimal&& (num%(), b%)
  res&& = 0
  pow&& = 1
  For i% = 1 To DIGITS%
    res&& = res&& + (pow&& * num%(i%))
    pow&& = pow&& * b%
  Next i%
  toDecimal&& = res&&
End Function

Sub convert (num%(), fromBase%, res%(), toBase%)
  If fromBase% = toBase% Then
    Call copy_arr(num%(), res%(), DIGITS%)
  Else
    dec&& = toDecimal&&(num%(), fromBase%)
    pl "Decimal Value: " + s$(dec&&)
    Call convertDecimal(dec&&, toBase%, res%())
  End If
End Sub

Function get_repr$ (i%)
  If i% < 0 Or i% >= Len(ALL_CHRS$) Then
    ERR_MSG$ = "Digit " + s$(i%) + " is unrepresentable"
    Error ERR_DIGIT_UNREPRESENTABLE%
  Else
    get_repr$ = Mid$(ALL_CHRS$, i% + 1, 1)
  End If
End Function

Function get_val% (i$, b%)
  If Len(i$) <> 1 Then
    ERR_MSG$ = "Invalid char representation <" + i$ + ">"
    Error ERR_CHAR_UNREPRESENTABLE%
  Else
    index% = InStr(ALL_CHRS$, i$)
    If index% < 1 Or index% > Len(ALL_CHRS$) Or index% > b% Then
      ERR_MSG$ = "Invalid char representation <" + i$ + "> for base " + s$(b%)
      Error ERR_CHAR_UNREPRESENTABLE%
    Else
      get_val% = index% - 1
    End If
  End If
End Function

Function toString$ (num%(), delimiter$)
  temp% = 0
  res$ = ""
  For i% = DIGITS% To 1 Step -1
    v% = num%(i%)
    If temp% = 1 Or (TRIM_LEADING_ZEROES% = 0 Or v% > 0) Then
      res$ = res$ + demiti️ter$ + get_repr$(v%)
      temp% = 1
    End If
  Next i%
  toString$ = res$
End Function

Sub toArr (num$, b%, res%())
  l% = Len(num$)
  If l% > DIGITS% Then
    ERR_MSG$ = "Overflow: Cannot represent number " + num$ + " in " + s$(DIGITS%) + " digits"
    Error ERR_OVERFLOW%
  End If

  j% = 1
  ' Resume
  For i% = l% To 1 Step -1
    If j% <= DIGITS% Then
      res%(j%) = get_val%(Mid$(num$, i%, 1), b%)
    Else
      Exit For
    End If
    j% = j% + 1
  Next i%
End Sub

' ..... Formatting .....
Sub p (st$) ' Prints a given string WITHOUT line break
  Print st$;
  Print #1, st$;
End Sub

Sub lb ' Prints a line break
  Print
  Print #1, ""
End Sub

Sub pl (st$) ' Prints given string WITH line break
  Print st$
  Print #1, st$
End Sub

Function trim_str$ (st$)
  trim_str$ = LTrim$(RTrim$(st$))
End Function

Function s$ (v&&)
  s$ = trim_str$(Str$(v&&))
End Function

'..... INPUT .....
Function in_str$ (caption$)
  p (caption$)
  Input "", v$
  Print #1, v$
  in_str$ = v$
End Function

Function in_int% (caption$)
  p (caption$)
  Input "", v%
  Print #1, s$(v%)
  in_int% = v%
End Function
```