

### P#1.1: Tonight or Tomorrow?

Code Word	Latin Transcription	Code Word	Latin Transcription
☰○□□□	2moro	◆□■△◆	tonyt
◆□○□%	tomoz	☰■)(yo 𐄂◆	2night
◆□○□□□	tomoro	☰■△◆	2nyt
◆□○□◆	tomrw	☰■~	2n8
◆□○	tom	◆□■~	ton8
◆□○□%	tomra	◆□■△◆%	tonytz
☰○□□	2mro	☰■)(◆m,	2nite
○□□%	mora	◆□■)(◆	tonit

Now decode the following Tweet (into Standard English):

□△◆ ■◆ ◆m □ 𐄂m 𐄂□𐄂)(■yo ◆)(■yo 𐄂)(■yo%

ryt nw we r decoding wingdingz

**Answer:** Right now we are decoding Wingdings.

#### Explanation:

The idea here is to look at the patterns in which the symbol occurs. For instance, Tomorrow has 3 o's in succession separated by consonants. The task also involves exploiting assumptions such as (a) first letter of the words are never deleted, though they might be substituted with a similar sounding letter of character.

#### Thinking Beyond:

Can you formalize the strategy you used to solve this problem in the form of an algorithm that will automatically discover the potential variants of a set of valid words from a list of contracted words? Note that like in this problem, the script used for the contracted words might be different from that used for the valid words (transliteration, eh?)

## P#1.2 SOUNDEX

(a)

*Allaway: A400, Anderson: A536, Ashcombe: A251, Buckingham: B252,*

*Chapman: C155, Colquhoun: C425, Evans: E152, Fairwright: F623,*

*Kingscott: K523, Lewis: L200, Littlejohns: L342, Stanmore: S356,*

*Stubbs: S312, Tocher: T260, Tonks: T520, Whytehead: W330.*

(b)

1. Leave the first letter in place.

2. Delete *h* and *w*.

3. Replace all consonant letters with digits (letters whose most common sounds are similar are grouped together):

*bpv(f):1      cgjkqs(xz):2    dt:3    l:4    mn:5    r:6*

4. Reduce any sequence of two or more identical digits to a single digit.

5. Delete all vowels (*a, e, i, o, u, y*).

6. Leave only the first three digits or add zeroes on the right to make the code one letter and three digits long.

(c)

*Ferguson: F622, Fitzgerald: F326, Hamnett: H530, Keefe: K100,*

*Maxwell: M240, Razey: R200, Shaw: S000, Upfield: U143.*

### Thinking Beyond:

1. What do you think is the rationale behind this encoding? More specifically:
  - a. Why the first letter remains intact?
  - b. Why the designer would have chosen to have a code-length of 4 (i.e., one char + 3 more digits), instead of 2 or 10?
2. SOUNDEX uses only 6 codes for the letters, and 0 for padding. What are the advantages or disadvantages of extending the codes to 9 (i.e., divide the consonants to 9 classes instead of 6), or 3? If you were to design letter classes with 9 or 3 sets, how would you do it?
3. Try to design a SOUNDEX like encoding scheme for your mother tongue (or any Indian language other than English).

## P#2.2: Māori Loanwords

<i>hāma</i> = hammer	<i>māti</i> = match-stick	<i>raina</i> = line	<i>tīhi</i> = cheese
<i>hāpa</i> = harp	<i>paipa</i> = pipe	<i>taraka</i> = truck	<i>tūru</i> = stool
<i>hū</i> = shoe	<i>piriti</i> = bridge	<i>terewhono</i> = telephone	<i>wāna</i> = swan
<i>hūtu</i> = suit	<i>pūnu</i> = spoon	<i>tiā</i> = jar	<i>whurutu</i> = fruit
<i>iniki</i> = ink	<i>pūtu</i> = boot	<i>tiaka</i> = jug	<i>wūru</i> = wool

*hekeretari* = secretary, *pirinihehe* = princess, *pirihimana* = policeman, *tiati* = judge

*Iharaira* = Israel, *Kiupa* = Cuba, *Peina* = Spain, *Tiamani* = Germany, *Tiapana* = Japan

beef = *pīwhi*, bull = *pūru*, cart = *kāta*, clock = *karaka*, lease = *rīhi*, meat = *mīti*, seal = *hīri*, street = *tiriti*, time = *taima*, watch = *wāta*

### Explanation:

Maori allows only CV (Consonant-Vowel) syllables. Therefore, Loan words ending with a consonant (e.g., swan) needs a vowel, and similarly, loan words that have multiple consonants (i.e., CCV) needs to be broken down into two syllables by inserting a vowel. E.g., truck = taraka

Maori doesn't have consonants such as b (replaced by p), sh (replaced by h), l (replaced by r), j (replaced by t).

In general, all languages have certain constraints on what combination of consonants and vowels are allowed in a valid word structure. As a result, transliteration of words from a foreign language always implies certain amount of phonological transformations. Usually, these rules are predictable.

### Thinking Beyond:

1. Can you write down an algorithm to convert an English loan word to Maori?
2. Imagine that you are transliterating Hindi loan words to English and vice versa. What are the sounds or sound combinations allowed in Hindi that are not allowed in English? What would be their most natural and closest transliteration into English? (E.g., English doesn't have the sound "d" as in "badA" [big]. How does this sound gets transliterated into English?). Repeat the same exercise in the reverse direction. For instance, Hindi doesn't have the English vowel "o" in "hot". [If you do not know Hindi, do the same exercise of any Indian language that you know].

## P#2.2 Transliterating Lepcha

Like all Brahmi derived scripts, Lepcha is an Abugida and is written from left to right.

- The CV (Consonant-vowel) syllables are written by writing down the consonant and the V is marked by a diacritic. E.g.,  $\text{ᳵ} (r) + \text{᳚} (aa) = \text{ᳶ} (raa)$  in Rai (the a here is a long aa like the a in “art”).
  - Some diacritics are written before the consonant (e.g.,  $\text{ᳵ} (l) + \text{᳚} (i) = \text{ᳶ} (li)$ )
  - Long vowels are represented by the diacritic  $\text{᳚}$ . But certain long vowels, like oo has special symbols  $\text{᳚}$  which is essentially  $\text{᳚}$  (therefore,  $\text{᳚}$  is short o)
  - The vowel short a (the sound of u in *fun*) is the default sound and is not represented by any diacritic. E.g., Magar =  $\text{ᳶ} (ma) + \text{᳚} (gar)$
- CVC syllables are expressed as the CV syllables along with a diacritic for the last consonant. E.g., ren =  $\text{ᳵ} (r) + \text{᳚} (e) + \text{᳚} (n) = \text{ᳶ}$  This is different from many other Brahmi derived scripts where the last consonant is attached to the next syllable. Only some consonants can occur in the syllable final position: *m, n, ng, r, p, t* etc.
- The diacritic for the syllable-final ng is an exception because it precedes the initial consonant. Thus, *mang* is written as  $\text{ᳶ} (ng) + \text{ᳶ} (m) + \text{᳚} (aa) = \text{ᳶ}$
- CCV syllables, where the only allowable consonants in the second position are r, y, or l (the problem provides example of only *r* in **Drendzongke**), are written with special diacritics as well. E.g.,  $\text{ᳶ} (d) + \text{᳚} (r) + \text{᳚} (e) + \text{᳚} (n) = \text{ᳶ}$  (dren)
- V syllables are written with a dummy consonant ( $\text{ᳶ}$ ) marker with the vowel diacritic. e.g., *i* =  $\text{ᳶ}$

The Lepcha consonants used in this problem are: n =  $\text{ᳶ}$ , p =  $\text{ᳶ}$ , l =  $\text{ᳶ}$ , ch =  $\text{ᳶ}$ , d =  $\text{ᳶ}$ , dz (j) =  $\text{ᳶ}$ , k =  $\text{ᳶ}$ , t =  $\text{ᳶ}$ , m =  $\text{ᳶ}$ , b =  $\text{ᳶ}$ , w =  $\text{ᳶ}$ , r =  $\text{ᳶ}$ , g =  $\text{ᳶ}$ , s =  $\text{ᳶ}$ , sh =  $\text{ᳶ}$ , h =  $\text{ᳶ}$ , dummy (used for vowels) =  $\text{ᳶ}$

The Lepcha vowel diacritics used in the problem are: aa =  $\text{᳚}$ , i =  $\text{᳚}$ , u =  $\text{᳚}$ , e =  $\text{᳚}$ , oo =  $\text{᳚}$

The Lepcha consonant diacritics (i.e., syllable final consonants) used in this problem are: p =  $\text{᳚}$ , n =  $\text{᳚}$ , ng =  $\text{᳚}$ , m =  $\text{᳚}$ , r =  $\text{᳚}$ , t =  $\text{᳚}$ ,

Assignment 1	Answer
Language Name	SIKKIMESE
Assignment 2	
ꨀꨁꨂꨃꨄ	Renjoongmu or Rendzoongmu
ꨀꨁꨂꨃꨄꨅ	Taamsaangmu
ꨀꨁꨂꨃꨄꨅꨆ	Hilaammu
ꨀꨁꨂꨃ	Promu
Assignment 3	
Transcribe <i>Kangchenjunga</i>	ꨀꨁꨂꨃꨄꨅꨆꨇꨈꨉꨊꨋꨌꨍꨎꨏꨐꨑꨒꨓꨔꨕꨖꨗꨘꨙꨚꨛꨜꨝꨞꨟꨠꨡꨢꨣꨤꨥꨦꨧꨨꨩꨪꨫꨬꨭꨮꨯꨰꨱꨲꨳꨴꨵꨶ꨷꨸꨹꨺꨻꨼꨽꨾꨿ꩀꩁꩂꩃꩄꩅꩆꩇꩈꩉꩊꩋꩌꩍ꩎꩏꩐꩑꩒꩓꩔꩕꩖꩗꩘꩙꩚꩛꩜꩝꩞꩟ꩠꩡꩢꩣꩤꩥꩦꩧꩨꩩꩪꩫꩬꩭꩮꩯꩰꩱꩲꩳꩴꩵꩶ꩷꩸꩹ꩺꩻꩼꩽꩾꩿꪀꪁꪂꪃꪄꪅꪆꪇꪈꪉꪊꪋꪌꪍꪎꪏꪐꪑꪒꪓꪔꪕꪖꪗꪘꪙꪚꪛꪜꪝꪞꪟꪠꪡꪢꪣꪤꪥꪦꪧꪨꪩꪪꪫꪬꪭꪮꪯꪰꪱꪴꪲꪳꪵꪶꪷꪸꪹꪺꪻꪼꪽꪾ꪿ꫀ꫁ꫂ꫃꫄꫅꫆꫇꫈꫉꫊꫋꫌꫍꫎꫏꫐꫑꫒꫓꫔꫕꫖꫗꫘꫙꫚ꫛꫜꫝ꫞꫟ꫠꫡꫢꫣꫤꫥꫦꫧꫨꫩꫪꫫꫬꫭꫮꫯ꫰꫱ꫲꫳꫴꫵ꫶꫷꫸꫹꫺꫻꫼꫽꫾꫿꬀ꬁꬂꬃꬄꬅꬆ꬇꬈ꬉꬊꬋꬌꬍꬎ꬏꬐ꬑꬒꬓꬔꬕꬖ꬗꬘꬙꬚꬛꬜꬝꬞꬟ꬠꬡꬢꬣꬤꬥꬦ꬧ꬨꬩꬪꬫꬬꬭꬮ꬯ꬰꬱꬲꬳꬴꬵꬶꬷꬸꬹꬺꬻꬼꬽꬾꬿꭀꭁꭂꭃꭄꭅꭆꭇꭈꭉꭊꭋꭌꭍꭎꭏꭐꭑꭒꭓꭔꭕꭖꭗꭘꭙꭚ꭛ꭜꭝꭞꭟꭠꭡꭢꭣꭤꭥꭦꭧꭨꭩ꭪꭫꭬꭭꭮꭯ꭰꭱꭲꭳꭴꭵꭶꭷꭸꭹꭺꭻꭼꭽꭾꭿꮀꮁꮂꮃꮄꮅꮆꮇꮈꮉꮊꮋꮌꮍꮎꮏꮐꮑꮒꮓꮔꮕꮖꮗꮘꮙꮚꮛꮜꮝꮞꮟꮠꮡꮢꮣꮤꮥꮦꮧꮨꮩꮪꮫꮬꮭꮮꮯꮰꮱꮲꮳꮴꮵꮶꮷꮸꮹꮺꮻꮼꮽꮾꮿꯀꯁꯂꯃꯄꯅꯆꯇꯈꯉꯊꯋꯌꯍꯎꯏꯐꯑꯒꯓꯔꯕꯖꯗꯘꯙꯚꯛꯜꯝꯞꯟꯠꯡꯢꯣꯤꯥꯦꯧꯨꯩꯪ꯫꯬꯭꯮꯯꯰꯱꯲꯳꯴꯵꯶꯷꯸꯹꯺꯻꯼꯽꯾꯿가각갂갃간갅갆갇갈갉갊갋갌갍갎갏감갑값갓갔강갖갗갘같갚갛개객갞갟갠갡갢갣갤갥갦갧갨갩갪갫갬갭갮갯갰갱갲갳갴갵갶갷갸갹갺갻갼갽갾갿걀걁걂걃걄걅걆걇걈걉걊걋걌걍걎걏걐걑걒걓걔걕걖걗걘걙걚걛걜걝걞걟걠걡걢걣걤걥걦걧걨걩걪걫걬걭걮걯거걱걲걳건걵걶걷걸걹걺걻걼걽걾걿검겁겂것겄겅겆겇겈겉겊겋게겍겎겏겐겑겒겓겔겕겖겗겘겙겚겛겜겝겞겟겠겡겢겣겤겥겦겧겨격겪겫견겭겮겯결겱겲겳겴겵겶겷겸겹겺겻겼경겾겿곀곁곂곃계곅곆곇곈곉곊곋곌곍곎곏곐곑곒곓곔곕곖곗곘곙곚곛곜곝곞곟고곡곢곣곤곥곦곧골곩곪곫곬곭곮곯곰곱곲곳곴공곶곷곸곹곺곻과곽곾곿관괁괂괃괄괅괆괇괈괉괊괋괌괍괎괏괐광괒괓괔괕괖괗괘괙괚괛괜괝괞괟괠괡괢괣괤괥괦괧괨괩괪괫괬괭괮괯괰괱괲괳괴괵괶괷괸괹괺괻괼괽괾괿굀굁굂굃굄굅굆굇굈굉굊굋굌굍굎굏교굑굒굓굔굕굖굗굘굙굚굛굜굝굞굟굠굡굢굣굤굥굦굧굨굩굪굫구국굮굯군굱굲굳굴굵굶굷굸굹굺굻굼굽굾굿궀궁궂궃궄궅궆궇궈궉궊궋권궍궎궏궐궑궒궓궔궕궖궗궘궙궚궛궜궝궞궟궠궡궢궣궤궥궦궧궨궩궪궫궬궭궮궯궰궱궲궳궴궵궶궷궸궹궺궻궼궽궾궿귀귁귂귃귄귅귆귇귈귉귊귋귌귍귎귏귐귑귒귓귔귕귖귗귘귙귚귛규귝귞귟균귡귢귣귤귥귦귧귨귩귪귫귬귭귮귯귰귱귲귳귴귵귶귷그극귺귻근귽귾귿글긁긂긃긄긅긆긇금급긊긋긌긍긎긏긐긑긒긓긔긕긖긗긘긙긚긛긜긝긞긟긠긡긢긣긤긥긦긧긨긩긪긫긬긭긮긯기긱긲긳긴긵긶긷길긹긺긻긼긽긾긿김깁깂깃깄깅깆깇깈깉깊깋까깍깎깏깐깑깒깓깔깕깖깗깘깙깚깛깜깝깞깟깠깡깢깣깤깥깦깧깨깩깪깫깬깭깮깯깰깱깲깳깴깵깶깷깸깹깺깻깼깽깾깿꺀꺁꺂꺃꺄꺅꺆꺇꺈꺉꺊꺋꺌꺍꺎꺏꺐꺑꺒꺓꺔꺕꺖꺗꺘꺙꺚꺛꺜꺝꺞꺟꺠꺡꺢꺣꺤꺥꺦꺧꺨꺩꺪꺫꺬꺭꺮꺯꺰꺱꺲꺳꺴꺵꺶꺷꺸꺹꺺꺻꺼꺽꺾꺿껀껁껂껃껄껅껆껇껈껉껊껋껌껍껎껏껐껑껒껓껔껕껖껗께껙껚껛껜껝껞껟껠껡껢껣껤껥껦껧껨껩껪껫껬껭껮껯껰껱껲껳껴껵껶껷껸껹껺껻껼껽껾껿꼀꼁꼂꼃꼄꼅꼆꼇꼈꼉꼊꼋꼌꼍꼎꼏꼐꼑꼒꼓꼔꼕꼖꼗꼘꼙꼚꼛꼜꼝꼞꼟꼠꼡꼢꼣꼤꼥꼦꼧꼨꼩꼪꼫꼬꼭꼮꼯꼰꼱꼲꼳꼴꼵꼶꼷꼸꼹꼺꼻꼼꼽꼾꼿꽀꽁꽂꽃꽄꽅꽆꽇꽈꽉꽊꽋꽌꽍꽎꽏꽐꽑꽒꽓꽔꽕꽖꽗꽘꽙꽚꽛꽜꽝꽞꽟꽠꽡꽢꽣꽤꽥꽦꽧꽨꽩꽪꽫꽬꽭꽮꽯꽰꽱꽲꽳꽴꽵꽶꽷꽸꽹꽺꽻꽼꽽꽾꽿꾀꾁꾂꾃꾄꾅꾆꾇꾈꾉꾊꾋꾌꾍꾎꾏꾐꾑꾒꾓꾔꾕꾖꾗꾘꾙꾚꾛꾜꾝꾞꾟꾠꾡꾢꾣꾤꾥꾦꾧꾨꾩꾪꾫꾬꾭꾮꾯꾰꾱꾲꾳꾴꾵꾶꾷꾸꾹꾺꾻꾼꾽꾾꾿꿀꿁꿂꿃꿄꿅꿆꿇꿈꿉꿊꿋꿌꿍꿎꿏꿐꿑꿒꿓꿔꿕꿖꿗꿘꿙꿚꿛꿜꿝꿞꿟꿠꿡꿢꿣꿤꿥꿦꿧꿨꿩꿪꿫꿬꿭꿮꿯꿰꿱꿲꿳꿴꿵꿶꿷꿸꿹꿺꿻꿼꿽꿾꿿

### Explanation:

Unlike English, which is an *alphabetic script*, Lepcha is a more complex script that is called a syllabary. Each syllable is written using a complex composition of the characters. Japanese is another example of syllabary. Other Indic scripts, such as Devanagari, is close to a syllabary, but have some small differences – such as the consonant sound at the end of the syllable (technically called coda) is shown as a part of the next akshara (e.g., the Hindi word काल has two akshars, but only one syllable phonologically /kal/. So an akshara is not an exact equivalent of a syllable.

Transliteration strongly depends on the type of scripts involved during the process. An important question one must consider is what should be treated as the minimal unit of the writing system (technically such a minimal unit is called a grapheme). While we consider letters as the minimal unit, but that is only useful in the case of alphabetic scripts. For other kinds of scripts, one must consider syllables or akshars as possible minimal units.

### Thinking Beyond:

Read more about the different kinds of writing systems ([https://en.wikipedia.org/wiki/Writing\\_system](https://en.wikipedia.org/wiki/Writing_system)) and think about what would be the ideal choice for the minimal unit while transliterating between two different kinds of scripts.

## **P#2.3 Switching ya Mixing?**

**2.1a** Dude I think u should try again caz ye [this] tera [your] fault nahi [not] hai [is]. ye [this] CBSE walo [people] ki [of] fault hai [is].

*Translation:* Dude I think you should try again because it is not your fault at all. This is CBSE folk's fault.

**Answer: Switching & mixing\***

Dude I think u should try again caz: [Matrix=En]

ye tera fault\* nahi hai. [Matrix = Hi]

ye CBSE walo ki fault\* hai. [Matrix = Hi]

**2.1b** Corruption to [grammatical particle] every level pe [at] hai [is] and its complete eradication possible nahin [is not].

*Translation:* Corruption is at every level and its complete eradication is not possible.

**Answer: Mixing, Matrix = Hi**

**2.1c** I had told you exams me [in] difficult questions aayenge [will come].

*Translation:* I had told you that there will be difficult questions in the exam.

**Answer: Switching & mixing\***

I had told you [Matrix = En]

Exams\* me difficult\* questions\* aayenge. [Matrix = Hi]

### **Thinking Beyond:**

Can you design a system to automatically classify a mixed language text into code-mixing or code-switching? What resources would you need?

What about automatic identification of the matrix?