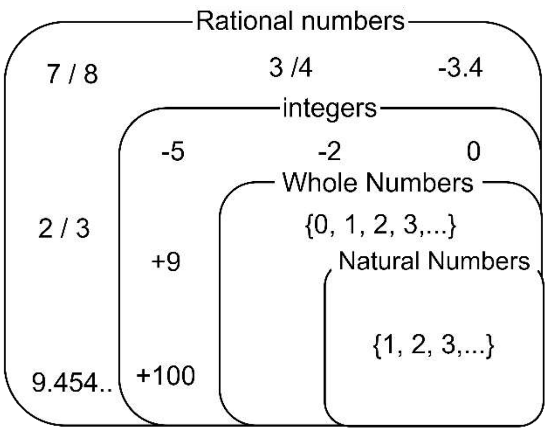
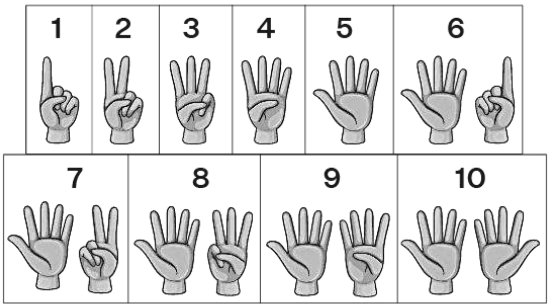
Contents

Number Systems
සංඛ්‍යා පද්ධති

A set of symbols which can be used to represent any finite value is known as a number system.
ඕනෑම පරිමිත වටිනාකමක් දැක්වීම සඳහා භාවිතා කල හැකි සංකේත කිහිපයක එකතුවක් සංඛ්‍යා පද්ධතියකි



Different number systems are used in different places. Numbers help count things, know how many, and show order.
විවිධ ස්ථානවල විවිධ සංඛ්‍යා පද්ධති භාවිතා වේ. ඉලක්කම් දේවල් ගණන් කිරීමට, කොපමණ ප්‍රමාණයක්ද යන්න දැනගැනීමට, සහ පිළිවෙල පෙන්වීමට උදවු කරයි.



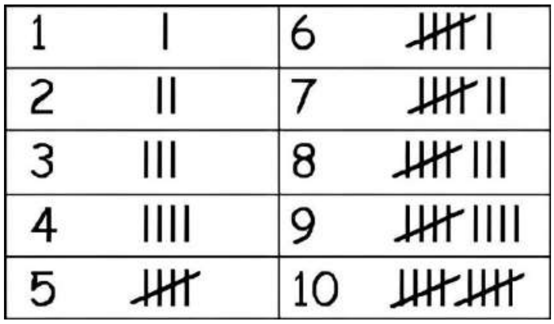
History of Number Systems
සංඛ්‍යා පද්ධති වල ඉතිහාසය

Number systems have a long history, helping people count, trade, and solve problems.
සංඛ්‍යා පද්ධති වලට දිගු ඉතිහාසයක් ඇති අතර ඒවා මගින් මිනිසුන්ට ගණන් කිරීමට, වෙළඳාම් කිරීමට සහ ගැටළු විසඳීමට උපකාර කරයි.

1. Tally Marks

One of the earliest methods of counting was using tally marks.
ගණන් කිරීමේ මුල්ම ක්‍රමයක් වූයේ tally marks භාවිතා කිරීමයි.

A simple method of counting objects or events using small vertical lines is known as Tally marks.
කුඩා සිරස් රේඛා භාවිතයෙන් වස්තූන් හෝ සිදුවීම් ගණනය කිරීමේ සරල ක්‍රමයක් ලෙස Tally marks හදුන්වයි.



Tally marks are useful because they are simple to understand and use, making them accessible to all.
Tally marks ප්‍රයෝජනවත් වන්නේ ඒවා තේරුම් ගැනීමට සහ භාවිතා කිරීමට සරල බැවින්, ඒවා සියල්ලන්ටම ප්‍රවේශ විය හැකි බැවිනි.

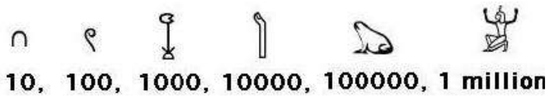
Complex calculations or special characters aren’t needed, allowing for quick and efficient counting.
ඉක්මන් සහ කාර්යක්ෂම ගණන් කිරීමට ඉඩ සලසන සංකීර්ණ ගණනය කිරීම් හෝ විශේෂ අක්ෂර අවශ්‍ය නොවේ.

Tally marks are still used today, especially in Some situation like,අදටත් විශේෂයෙන්ම සමහර අවස්ථා වලදී tally සලකුණු භාවිතා වේ.

2. Egyptian Numerals
ඊජිප්තියානු සංඛ්‍යා

Egyptians developed a system using hieroglyphs to represent numbers.
ඊජිප්තුවරුන් සංඛ්‍යා නිරූපණය කිරීම සඳහා හයිරොග්ලිෆ් භාවිතා කරන පද්ධතියක් නිර්මාණය කර ඇත.

They had separate symbols for units, tens, hundreds, thousands, and so on.
ඔවුන්ට ඒකක, දස, සිය, දහස්, යනාදී වශයෙන් වෙනම සංකේත තිබුණි.



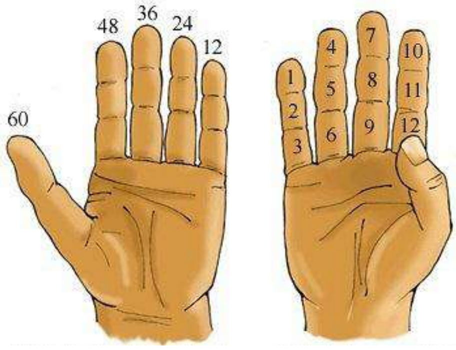
Example:
The number 2,432 would be written as,
2,432 අංකය මෙසේ ලියා ඇත.

2432 = 

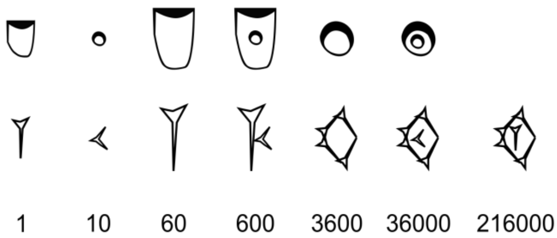
2x1000 4x100 3x10 2x 1

3. Babylonian Base 60
බැබිලෝනියානු 60 පාදය

The Babylonians used a base-60 system (around 1900-1600 BC), which influenced our modern timekeeping (60 seconds in a minute, 60 minutes in an hour and 360 degrees in a circle).
බැබිලෝනියානුවන් අපගේ නවීන කාලසටහනට (විනාඩියකට තත්පර 60, පැයකට මිනිත්තු 60 සහ රවුමක අංශක 360) බලපෑ 60 පාදයේ පද්ධතියක් (ක්‍රි.පූ. 1900-1600 පමණ) භාවිතා කර ඇත.



It's believed that the Babylonians chose base-60 because it is highly divisible by many numbers, including 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, and 60 itself. This made calculations and fractions easier to work with.
2, 3, 4, 5, 6, 10, 12, 15, 20, 30 සහ 60 ඇතුළුව බොහෝ සංඛ්‍යාවලින් බෙහෙවින් බෙදිය හැකි නිසා බබිලෝනිවරුන් 60 පාදය තෝරා ගත් බව විශ්වාස කෙරේ. මෙය ගණනය කිරීම් සහ භාග සමඟ වැඩ කිරීමට පහසු විය.



Example:
The number 123 in the decimal system would be written as 2 sixties and 3 ones in the Babylonian system.
දශමය ක්‍රමයේ අංක 123 ලියා ඇත්තේ හැටේ 2ක් සහ බැබිලෝනියානු ක්‍රමයේ එකක් ලෙසය.
This would be represented as,
මෙය නිරූපනය වනු ඇත්තේ,

123 = 

2x60 3x1

The Babylonian base 60 system had a profound influence on our modern-day timekeeping system.
බැබිලෝනියානු 60 පාදයේ ක්‍රමය අපගේ නූතන කාල සටහන් පද්ධතියට ප්‍රබල බලපෑමක් ඇති කළේය.

We still divide hours into 60 minutes and minutes into 60 seconds, and the division of the circle into 360 degrees can be traced back to Babylonian astronomy.
තවමත් පැය විනාඩි 60 කට මිනිත්තුව තත්පර 60 කට සහ වෘත්තයක් අංශක 360 කට බෙදීම බැබිලෝනියානු තාරකා විද්‍යාවෙන් සොයාගත හැකිය.

Although they are not commonly used today for calculations, they still appear in certain contexts, such as clocks, book chapters, and movie sequels.
අද ඒවා ගණනය කිරීම් සඳහා බහුලව භාවිතා නොවුනත්, ඒවා තවමත් ඔරලෝසු, පොත් පරිච්ඡේද සහ චිත්‍රපට අනුක්‍රමික වැනි ඇතැම් සන්දර්භවල දක්නට ලැබේ.

4. Roman Numerals
රෝමානු ඉලක්කම්

Roman numerals are a system of numerical notation used by the ancient Romans.
රෝම ඉලක්කම් යනු පුරාණ රෝමවරුන් විසින් භාවිතා කරන ලද සංඛ්‍යාත්මක අංකන පද්ධතියකි.

Instead of using digits like 1, 2, 3, they used specific letters to represent numbers.
1, 2, 3 වැනි ඉලක්කම් භාවිතා කිරීම වෙනුවට, ඔවුන් ඉලක්කම් නියෝජනය කිරීමට විශේෂිත අකුරු භාවිතා කර ඇත.

How Roman Numerals Work
රෝම ඉලක්කම් ක්‍රියා කරන ආකාරය

Roman numerals are written by combining these letters. Add the values together, but if a smaller numeral is placed before a larger one, subtract the smaller value.
රෝම ඉලක්කම් ලියා ඇත්තේ මෙම අකුරු එකතු කිරීමෙනි. අගයන් එකට එකතු කරයි, නමුත් විශාල සංඛ්‍යාවක් ඉදිරියේ කුඩා සංඛ්‍යාවක් තැබුවහොත්, කුඩා අගය අඩු කළ යුතුය.

Counting 1 to 100 from Roman Number System
රෝම ඉලක්කම් වලින් 1 සිට 100 දක්වා

1 = I	11 = XI	21 = XXI	31 = XXXI	41 = XLI	51 = LI	61 = LXI	71 = LXXI	81 = LXXXI	91 = XCI
2 = II	12 = XII	22 = XXII	32 = XXXII	42 = XLII	52 = LII	62 = LXII	72 = LXXII	82 = LXXXII	92 = XCII
3 = III	13 = XIII	23 = XXIII	33 = XXXIII	43 = XLIII	53 = LIII	63 = LXIII	73 = LXXIII	83 = LXXXIII	93 = XCIII
4 = IV	14 = XIV	24 = XXIV	34 = XXXIV	44 = XLIV	54 = LIV	64 = LXIV	74 = LXXIV	84 = LXXXIV	94 = XCIV
5 = V	15 = XV	25 = XXV	35 = XXXV	45 = XLV	55 = LV	65 = LXV	75 = LXXV	85 = LXXXV	95 = XCV
6 = VI	16 = XVI	26 = XXVI	36 = XXXVI	46 = XLVI	56 = LVI	66 = LXVI	76 = LXXVI	86 = LXXXVI	96 = XCVI
7 = VII	17 = XVII	27 = XXVII	37 = XXXVII	47 = XLVII	57 = LVII	67 = LXVII	77 = LXXVII	87 = LXXXVII	97 = XCVII
8 = VIII	18 = XVIII	28 = XXVIII	38 = XXXVIII	48 = XLVIII	58 = LVIII	68 = LXVIII	78 = LXXVIII	88 = LXXXVIII	98 = XCVIII
9 = IX	19 = XIX	29 = XXIX	39 = XXXIX	49 = XLIX	59 = LIX	69 = LXIX	79 = LXXIX	89 = LXXXIX	99 = XCIX
10 = X	20 = XX	30 = XXX	40 = XL	50 = L	60 = LX	70 = LXX	80 = LXXX	90 = XC	100 = C

Each symbol in Roman numerals represents a specific value:
රෝම ඉලක්කම්වල සෑම සංකේතයක්ම නිශ්චිත අගයක් නියෝජනය කරයි:

1	5	10	50	100	500	1000
I	V	X	L	C	D	M

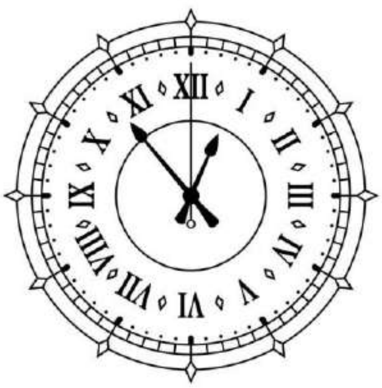
Number	Roman Numeral	Number	Roman Numeral
5,000	$\overline{V}$	5,000,000	$\overline{\overline{V}}$
10,000	$\overline{X}$	10,000,000	$\overline{\overline{X}}$
50,000	$\overline{L}$	50,000,000	$\overline{\overline{L}}$
100,000	$\overline{C}$	100,000,000	$\overline{\overline{C}}$
500,000	$\overline{D}$	500,000,000	$\overline{\overline{D}}$
1,000,000	$\overline{M}$	1,000,000,000	$\overline{\overline{M}}$

Uses of Roman numerals
රෝම ඉලක්කම්වල භාවිතය

Roman numerals are used on clock faces (e.g., III for 3, VI for 6).
ඔරලෝසු මුහුණුවල රෝම ඉලක්කම් භාවිතා වේ (උදා: 3 සඳහා III, 6 සඳහා VI).

In books or movies, they are used to denote chapters, volumes, or sequels (e.g., Chapter IX or Super Bowl L).
පොත්වල හෝ චිත්‍රපටවල, පරිච්ඡේද, volumes හෝ අනුප්‍රාප්තික (උදා. IX වන පරිච්ඡේදය හෝ Super Bowl L) දැක්වීමට භාවිතා කරයි.

Roman numerals are also used in monarch names (e.g., King Parakramabhahu II, Queen Elizabeth II).
රෝම ඉලක්කම් රාජාණ්ඩු නාමවල ද භාවිතා වේ (උදා: II පරාක්‍රමබාහු රජු, II එලිසබෙත් රැජින).



5. Greek Alphabet
ග්‍රීක හෝඩිය

Greeks used their alphabet to represent numbers, assigning different letters to different values.
ග්‍රීක ජාතිකයන් සංඛ්‍යා නියෝජනය කිරීමට ඔවුන්ගේ හෝඩිය භාවිතා කළ අතර, විවිධ අගයන් සඳහා විවිධ අකුරු ලබාදී ඇත.

It consists of 24 letters, each with an uppercase and lowercase form.
එය අකුරු 24 කින් සමන්විත වන අතර ඒ සෑම එකකටම uppercase සහ lowercase ආකාර ඇත.

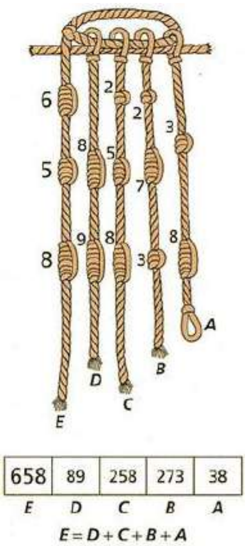
	Units	Tens	Hundreds
1	α alpha	ι iota	ρ rho
2	β beta	κ kappa	σ sigma
3	γ gamma	λ lambda	τ tau
4	δ delta	μ mu	υ upsilon
5	ε epsilon	ν nu	φ phi
6	ϝ digamma	ξ xi	χ chi
7	ζ zeta	ο omicron	ψ psi
8	η eta	π pi	ω omega
9	θ theta	ϙ koppa	ϡ sampi

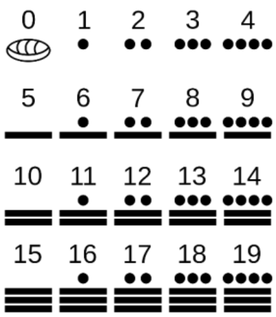
The concept of zero and the decimal system (base-10) originated in ancient India with the Indian decimal system.
ශුන්‍ය සංකල්පය සහ දශම ක්‍රමය (10-පාදය) යන සංකල්පය ඉපැරණි ඉන්දියාවේ ඉන්දියානු දශම ක්‍රමය සමඟ ආරම්භ විය.

The system we use today, often referred to as Hindu Arabic numerals, was developed from the Indian numeral system.
බොහෝ විට හින්දු අරාබි ඉලක්කම් ලෙස හැඳින්වෙන අද අප භාවිතා කරන ක්‍රමය, ඉන්දියානු සංඛ්‍යා ක්‍රමයෙන් වර්ධනය විය.

Brahmi		—	=	≡	+	𑁖	𑁗	𑁘	𑁙	𑁚
Hindu	०	१	२	३	४	५	६	७	८	९
Arabic	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
Medieval	o	I	2	3	ጸ	ς	6	ʌ	8	9
Modern	0	1	2	3	4	5	6	7	8	9

The quipu system of knotted strings used by the Inca or the base-20 system of the Mayans were also indigenous systems.
ඉන්කාවරුන් විසින් භාවිතා කරන ලද ගැට සහිත නූල් පද්ධතිය හෝ මායාවරුන්ගේ 20 පාදයේ සංඛ්‍යා පද්ධතිය ද දේශීය සංඛ්‍යා පද්ධති විය.





DID YOU KNOW?

The Greek alphabet is primarily used to represent the Greek language, one of the oldest languages in Europe.
යුරෝපයේ පැරණිතම භාෂාවක් වන ග්‍රීක භාෂාව නියෝජනය කිරීමට ග්‍රීක හෝඩිය මූලික වශයෙන් භාවිතා වේ.

Many mathematical and scientific symbols are derived from Greek letters. For example, pi (π), delta (Δ), and theta (Θ) are commonly used in mathematics and physics.
බොහෝ ගණිතමය සහ විද්‍යාත්මක සංකේත ග්‍රීක අක්ෂර වලින් ව්‍යුත්පන්න වී ඇත. උදාහරණයක් ලෙස, pi (π), ඩෙල්ටා (Δ) සහ තීටා (Θ) ගණිතයේ සහ භෞතික විද්‍යාවේ බහුලව භාවිතා වේ.

Greek letters are used in various fields for naming and classifying things. For instance, in astronomy, stars are often designated by Greek letters followed by the constellation name.
ග්‍රීක අකුරු විවිධ ක්ෂේත්‍රවල දේවල් නම් කිරීම සහ වර්ග කිරීම සඳහා යොදා ගනී. නිදසුනක් වශයෙන්, තාරකා විද්‍යාවේදී, තාරකා බොහෝ විට ග්‍රීක අක්ෂරවලින් පසුව තාරකා මණ්ඩලයේ නමෙන් නම් කරනු ලැබේ.

Throughout history, the development and adoption of various number systems have been influenced by cultural, economic, and technological factors.
ඉතිහාසය පුරාවට විවිධ සංඛ්‍යා පද්ධතිවල සංවර්ධනය සහ අනුගත වීම මත සංස්කෘතික, ආර්ථික සහ තාක්ෂණික සාදක බලපා ඇත.

The need for trade, astronomy, record-keeping, and later, computing, drove the evolution and spread of these systems.
වෙළඳාම, තාරකා විද්‍යාව, වාර්තා තබා ගැනීම සහ පසුව පරිගණකකරණය සඳහා වූ අවශ්‍යතාවය මෙම සංඛ්‍යා පද්ධතිවල පරිණාමය හා ව්‍යාප්තියට හේතු විය.

Number systems can generally be categorized into two main types
සංඛ්‍යා පද්ධති සාමාන්‍යයෙන් ප්‍රධාන වර්ග දෙකකට වර්ග කළ හැක

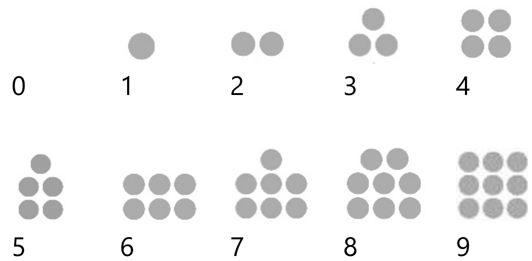
1. Positional Number Systems
ස්ථානීය වටිනාකමක් සහිත සංඛ්‍යා පද්ධති

The value of each symbol depends on its position within the number.
එක් එක් සංකේතයේ අගය අංකය තුළ එහි පිහිටීම මත රඳා පවතී.

The number of distinct symbols or digits in the system is determined by its base.
පද්ධතියේ ඇති වෙනස් සංකේත හෝ ඉලක්කම් ගණන එහි පාදය අනුව තීරණය වේ.

Decimal (base-10) - Digits 0-9
Binary (base-2) - Digits 0-1

The base of a positional number system determines the number of unique digits it uses.
ස්ථානීය වටිනාකමක් සහිත සංඛ්‍යා පද්ධතියක පාදය එය භාවිතා කරන අනන්‍ය ඉලක්කම් ගණන තීරණය කරයි.



Example:

100	10	1	100	10	1
10^2	10^1	10^0	10^2	10^1	10^0
7	**5**	**2**	**2**	**7**	**5**
7x100	5x10	2x1	2x100	7x10	5x1
700	50	2	200	70	5

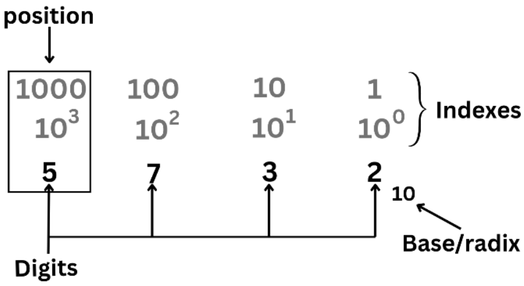
700+50+2 200+70+5

752_{10} > 275_{10}

The value of each position is the power of the base. එක් එක් ස්ථානයේ වටිනාකම පාදයේ බලයකි.

Here comes the zero concept when the value of any place is null.
යම් ස්ථානයක කිසිදු අගයක් නැතිවිට පැමිණෙන 0 සංකල්පයක් පවතී

Positional systems are more compact. Bigger values can be easily represented.
ස්ථානීය පද්ධති වඩාත් සංයුක්ත වේ. විශාල අගයන් පහසුවෙන් නිරූපණය කල හැකිය.



They have become the standard, especially with the advent of electronic computing, due to their computational efficiency.
විශේෂයෙන්ම ඉලෙක්ට්‍රොනික පරිගණනයේ පැමිණීමත් සමඟම මෙම වර්ගයේ සංඛ්‍යා පද්ධති වල කාර්යක්ෂමතාවය නිසා ඒවා වර්ථමාන භාවිතය බවට පත්ව ඇත.

Arithmetic operations like addition, subtraction, multiplication, and division are more straightforward in positional systems due to their structured nature.
එකතු කිරීම, අඩු කිරීම, ගුණ කිරීම සහ බෙදීම වැනි අංක ගණිත මෙහෙයුම් මෙම වර්ගයේ සංඛ්‍යා පද්ධති වල ව්‍යුහාත්මක ස්වභාවය නිසා වඩාත් සරල ය.

2. Non-positional number system
ස්ථානීය වටිනාකමක් රහිත සංඛ්‍යා පද්ධති

Non-positional number systems are systems where the value of a digit does not depend on its position within the number.
ස්ථානීය වටිනාකමක් රහිත සංඛ්‍යා පද්ධති යනු සංඛ්‍යාංකයක අගය අංකය තුළ පිහිටීම මත රඳා නොපවතින පද්ධති වේ.

Do not have a base. It has a set of different symbols with exact values.
පාදයක් නොපවතී. නිශ්චිත අගයන් සහිත විවිධ සංකේත සමූහයක් පවතී.

The value of a number is the sum of the values of its symbols, without regard to their order or position.
සංඛ්‍යාවක අගය යනු එහි සංකේතවල එකතුවයි. අනුපිළිවෙල හෝ පිහිටීම නොසලකයි.

Example:
Roman numerals, Egyptian numerals, Mayan numerals
රෝම ඉලක්කම්, ඊජිප්තු ඉලක්කම්, මායාඉලක්කම්

Key feature of Non-Positional Number Systems
ස්ථානීය නොවන අංක පද්ධතිවල ප්‍රධාන ලක්ෂණය

There's no concept of zero.
ශුන්‍ය සංකල්පයක් නොපවතී.

Has simplicity in representation for basic counting.
මූලික ගණන් කිරීම සඳහා සරල නිරූපණයක් පවතී.

Not compact. Bigger values cannot be easily represented.
සංයුක්ත නොවේ. විශාල අගයන් පහසුවෙන් නිරූපණය කළ නොහැක.

In non-positional systems, Arithmetic operations like addition, subtraction, multiplication, and division is more complex.
ස්ථානීය නොවන පද්ධතිවල, එකතු කිරීම, අඩු කිරීම, ගුණ කිරීම සහ බෙදීම වැනි අංක ගණිතමය මෙහෙයුම් වඩාත් සංකීර්ණ වේ.

Symbols represented fixed values for counting days, months, and years.
සංකේත දින, මාස සහ වසර ගණන් කිරීම සඳහා ස්ථාවර අගයන් නියෝජනය කරයි.

Roman numerals are often used in documents to number sections, chapters, and clause.
රෝමානු ඉලක්කම් බොහෝ විට ලේඛනවල කොටස්, පරිච්ඡේද සහ වගන්ති අංක කිරීමට භාවිතා කරයි.

Roman numerals are often used in titles of films and works of art, especially for sequels.
චිත්‍රපටවල හෝ කලා කෘතිවල, විශේෂයෙන් අනුප්‍රාප්තිකවල රෝමානු ඉලක්කම් භාවිතා වේ.

Used in ancient civilizations like the Egyptians and Romans, used non-positional systems to record quantities and dates.
ඊජිප්තුවරුන් සහ රෝමවරුන් වැනි පුරාණ ශිෂ්ටාචාරවල භාවිතා කරන ලද, ප්‍රමාණ සහ දිනයන් වාර්තා කිරීම සඳහා ස්ථානීය නොවන පද්ධති භාවිතා කරන ලදී.

Comparison of Positional Number Systems and Non-Positional Number Systems
ස්ථානීය වටිනාකමක් සහිත සහ ස්ථානීය වටිනාකමක් රහිත සංඛ්‍යා පද්ධති සංසන්දනය කිරීම

Feature **මූලාංග**	**Positional Number System** **ස්ථානීය වටිනාකමක් සහිත සංඛ්‍යා පද්ධති**	**Non-Positional Number System** **ස්ථානීය වටිනාකමක් රහිත සංඛ්‍යා පද්ධති**
Value of digits ඉලක්කම්වල අගය	Depends on position (place value) ස්ථානය (ස්ථාන අගය) මත රඳා පවතී	Constant, regardless of position ස්ථාවරය, ස්ථානය මත රඳා නොපවතී
Base (radix) පාදය	Requires a base, such as 10 (decimal) or 2 (binary) 10 (දශම) හෝ 2 (ද්විමය) වැනි පාදයක් අවශ්‍ය වේ	No base is used පාදයක් භාවිතා නොවේ
Symbols used භාවිතා කරන සංකේත	Limited symbols based on the base පාදය මත පදනම් වූ සීමිත සංකේත	Infinite number of symbols සංකේත අනන්ත ගණනක්
Ease of calculation ගණනය කිරීමේ පහසුව	Easier for arithmetic operations අංක ගණිතමය මෙහෙයුම් සඳහා වඩාත් පහසු වේ	More complex calculations වඩාත් සංකීර්ණ ගණනය කිරීම්
Zero representation ශූන්‍ය නියෝජනය	Includes zero බිංදුව ඇතුළත් වේ	Usually does not include zero සාමාන්‍යයෙන් බිංදුව ඇතුළත් නොවේ
Large numbers විශාල සංඛ්‍යා	Easy to represent නියෝජනය කිරීමට පහසුය	Difficult to represent නියෝජනය කිරීමට අපහසුය
Usage භාවිතය	Common in modern systems (computers, technology) නවීන පද්ධතිවල බහුලව දක්නට ලැබේ (පරිගණක, තාක්ෂණය)	Mostly historical (e.g., Roman numerals) බොහෝ දුරට පෞරාණික වේ (උදා: රෝම ඉලක්කම්)

Common number system
සාමාන්‍ය සංඛ්‍යා පද්ධති

Number system **සංඛ්‍යා පද්ධතිය**	**Base** **පාදය**	**Number sequence** **සංඛ්‍යා අනුක්‍රමය**	**Use by** **පාවිච්චියට ගන්නා ලද්දේ**	
			Man **මිනිසා**	**Computer** **පරිගණකය**
Decimal	10	0, 1, 2, 3, …….,9	Yes	No
Binary	2	0,1	No	Yes
Octal	8	0, 1, 2, ….., 7	No	No
Hexa-decimal	16	0, 1, 2, …..,9, A, B, ….., F	No	Yes
:	:	:	:	:
Base 64	64	[0-9] + [A-Z] + [a-z]+[ ] [ ]	No	Yes
Base n	n	0,1,2, ……, (n-1)	No	Yes

YouTube video IDs are encoded using a Base-64 system.
YouTube වීඩියෝ 64 පාදයේ සංඛ්‍යා පද්ධතියක් භාවිතයෙන් කේතනය කර ඇත.

←— Video ID —→
https://www.youtube.com/watch?v=dQw4w9WgXcQ

This base includes 64 characters, which are:
මෙම පාදයට අක්ෂර 64 ක් ඇතුළත් වේ, ඒවා නම්:

- Uppercase letters: A-Z (26 characters)
- Lowercase letters: a-z (26 characters)
- Digits: 0-9 (10 characters)
- Special characters: "-" and "_" (2 characters)

With these 64 characters, unique 1-character video IDs can be generated, representing an enormous number of combinations to support a vast video library.
මෙම අක්ෂර 64 සමඟින්, අතිවිශාල වීඩියෝ සමූහයකට සහය වීම සඳහා අතිවිශාල සංයෝජන සංඛ්‍යාවක් නියෝජනය කරමින්, අනන්‍ය අනුලක්ෂණයක් වීඩියෝ ID එකක් ජනනය කළ හැක.

By using this Base-64 system, a more efficient use of space is provided compared to decimal (Base-10) or hexadecimal (Base-16), ensuring that each video on the platform is uniquely identified.
මෙම 64 පාදයේ සංඛ්‍යා පද්ධතිය භාවිතා කිරීමෙන්, දශම (10 පාදය) හෝ ෂඩ් දශමය (16 පාදය) හා සසඳන විට වේදිකාවේ සෑම වීඩියෝවක්ම අනන්‍ය ලෙස හඳුනා ගන්නා බව සහතික කරමින් වඩා වැඩි ඉඩ කාර්යක්ෂමතාවක් සපයනු ලැබේ

1. Decimal Number System
දශමය සංඛ්‍යා පද්ධතිය

The decimal system is a base 10 numeral system, meaning it uses ten digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
දශමය පද්ධතිය යනු 10 පාද සංඛ්‍යා පද්ධතියකි, එනම් එය ඉලක්කම් දහයක් 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 භාවිතා කරයි.

The number after 9 is represented by increasing the next higher place value and resetting the current position to 0 (e.g., 9 + 1 = 10).
9 න් පසු අංකය නිරූපණය කරනු ලබන්නේ ඊළඟ ඉහළ ස්ථාන අගය වැඩි කිරීම සහ වත්මන් ස්ථානය 0 (උදා: 9 + 1 = 10) වෙත නැවත සැකසීමෙනි.

Each digit in a decimal number has a specific place value based on its position. These place values are powers of 10.
දශම සංඛ්‍යාවක සෑම ඉලක්කමකටම එහි පිහිටීම අනුව නිශ්චිත ස්ථාන අගයක් ඇත. මෙම ස්ථාන අගයන් 10 බල වේ.

The decimal system is the most common numeral system used in everyday life for counting, arithmetic, and commerce because it aligns with our ten fingers, making it intuitive and practical.
දශම ක්‍රමය යනු ගණන් කිරීම, අංක ගණිතය සහ වාණිජ්‍යය සඳහා එදිනෙදා ජීවිතයේ භාවිතා වන වඩාත් සුලභ සංඛ්‍යා පද්ධතියයි, මන්ද එය අපගේ ඇඟිලි දහය සමඟ සමපාත වන අතර එය බුද්ධිමය සහ ප්‍රායෝගික වේ.

Why are decimal numbers used?
දශමය සංඛ්‍යා භාවිතා කරන්නේ ඇයි?

The decimal system fits well with humans, as we have ten fingers. This makes counting and simple math, like adding and subtracting, easier and more intuitive.
ඇඟිලි දහයක් ඇති බැවින් දශම ක්‍රමය මිනිසුන්ට හොඳින් ගැලපේ. මෙය ගණන් කිරීම සහ එකතු කිරීම සහ අඩු කිරීම වැනි සරල ගණිතය, පහසු සහ වඩාත් අවබෝධාත්මක කරයි.

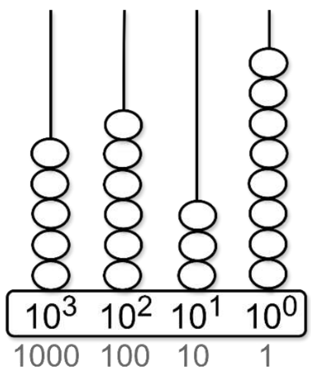
Decimal numbers are simple to learn and use. This system is common in education and daily tasks, like finance and measurement.
දශම සංඛ්‍යා ඉගෙනීමට සහ භාවිතා කිරීමට සරල වේ. මෙම ක්‍රමය අධ්‍යාපනයේ සහ මූල්‍ය සහ මිනුම් වැනි දෛනික කාර්යයන්හි පොදු වේ.

Basic operations, such as addition, subtraction, multiplication, and division, are easier in base-10. These are already familiar to most people, making calculations faster.
එකතු කිරීම, අඩු කිරීම, ගුණ කිරීම සහ බෙදීම වැනි මූලික මෙහෙයුම් 10 පාදයේ පහසු වේ. මේවා දැනටමත් බොහෝ මිනිසුන්ට හුරුපුරුදුය, එමගින් ගණනය කිරීම් වේගවත් වේ.

Decimal numbers work well with scientific notation, helping to represent very large or small numbers. This makes calculations easier for science and engineering.
දශම සංඛ්‍යා විද්‍යාත්මක අංකනය සමඟ හොඳින් ක්‍රියා කරයි, ඉතා විශාල හෝ කුඩා සංඛ්‍යා නියෝජනය කිරීමට උපකාරී වේ. මෙය විද්‍යාව සහ ඉංජිනේරු විද්‍යාව සඳහා ගණනය කිරීම් පහසු කරයි.

1.23×10^6 is used to represent 1,230,000.
1,230,000 නියෝජනය කිරීමට 1.23×10^6 භාවිතා වේ.

Let's consider the decimal number 5638_{10}.
5638_{10} යන දශමය සංඛ්‍යාව සලකමු.



Addition of two base 10 numbers
10 පාදයේ සංඛ්‍යා දෙකක් එකතු කිරීම

Example 01:
= $1987_{10} + 588_{10}$

1000	100	10	1
10^3	10^2	10^1	10^0
1	1	1	
1	9	8	7_{10}
+	5	8	8_{10}
2	15	17	15
	-10	-10	-10
2	5	7	5_{10}

Substraction of two base 10 numbers
10 පාදයේ සංඛ්‍යා දෙකක් අඩු කිරීම

Example 01:
= $1588_{10} – 979_{10}$

1000	100	10	1
10^3	10^2	10^1	10^0
	10		10
0 $\cancel{1}$	5	7 $\cancel{8}$	8_{10}
-	9	7	9_{10}
	6	0	9_{10}

2. Binary Number System
ද්විමය සංඛ්‍යා පද්ධතිය

The binary number system is a base-2 numeral system that uses only two symbols 0 and 1.
ද්විමය සංඛ්‍යා පද්ධතිය යනු 0 සහ 1 යන සංකේත දෙකක් පමණක් භාවිතා කරන පාද 2 සංඛ්‍යා පද්ධතියකි.

Each digit in a binary number is called a bit (binary digit).
ද්විමය සංඛ්‍යාවක සෑම ඉලක්කමක්ම බිටුවක් (ද්විමය ඉලක්කම්) ලෙස හැඳින්වේ.

Each position in a binary number has a place value that is a power of 2.
ද්විමය සංඛ්‍යාවක සෑම ස්ථානයකම ස්ථාන අගයක් ඇති අතර එය 2 බලයකි.

The binary number system is crucial for computers, which use binary to process data and execute instructions.
දත්ත සැකසීමට සහ උපදෙස් ක්‍රියාත්මක කිරීමට ද්විමය භාවිතා කරන පරිගණක සඳහා ද්විමය සංඛ්‍යා පද්ධතිය ඉතා වැදගත් වේ.

Digital circuits rely on binary states (on/off) to perform operations.
අංකිත පරිපථ මෙහෙයුම් සිදු කිරීම සඳහා ද්විමය තත්වයන් (on/off) මත රඳා පවතී.

It is the fundamental language of computers and digital systems because it corresponds directly to the two-state nature of electronic circuits, which can represent off (0) or on (1) states.
එය පරිගණක සහ අංකිත පද්ධතිවල මූලික භාෂාව වන්නේ එය ඉලෙක්ට්‍රොනික පරිපථවල ඕෆ් (0) හෝ ඔන් (1) තත්වයන් නියෝජනය කළ හැකි ද්විත්ව ස්වභාවයට සෘජුවම අනුරූප වන බැවිනි.

Binary numbers can be converted to other numeral systems, such as decimal (base-10) or hexadecimal (base-16), for ease of understanding and representation.
ද්විමය සංඛ්‍යා දශම (පාදය 10) හෝ ෂඩ්දශමය (පාදය 16) වැනි අනෙකුත් සංඛ්‍යා පද්ධතිවලට පරිවර්තනය කළ හැකි අතර, එය අවබෝධ කර ගැනීමේ සහ නිරූපණයේ පහසුව සඳහා වේ.

Binary numbers can be long and cumbersome, so they are often grouped into sets of four bits (nibbles) and expressed in hexadecimal for simplicity.
ද්විමය සංඛ්‍යා දිගු හා අපහසු විය හැක, එබැවින් ඒවා බොහෝ විට බිටු හතරක (නිබල්) කට්ටලවලට කාණ්ඩ කර ඇති අතර සරල බව සඳහා ෂඩ්දශමය ආකාරයෙන් ප්‍රකාශ වේ.

Why are binary numbers used?
ද්විමය සංඛ්‍යා භාවිතා කරන්නේ ඇයි ?

Using two states reduces errors common in more complex systems, making binary systems more reliable in digital electronics.
අවස්ථා දෙකක් භාවිතා කිරීම වඩාත් සංකීර්ණ පද්ධතිවල පොදු දෝෂ අවම කරයි, ද්විමය පද්ධති අංකිත ඉලෙක්ට්‍රොනික උපකරණවල වඩාත් විශ්වාසදායක කරයි.

Binary represents large numbers and complex data using fewer bits than systems like decimal.
Binary මගින් දශම වැනි පද්ධති වලට වඩා අඩු බිටු භාවිතා කරන විශාල සංඛ්‍යා සහ සංකීර්ණ දත්ත නියෝජනය කරයි.

It also standardizes data storage and processing in computers, simplifying the management of various information types (numbers, text, images, etc.).
එය පරිගණක තුළ දත්ත ගබඩා කිරීම සහ සැකසීම, විවිධ තොරතුරු වර්ග (අංක, පෙළ, රූප, ආදිය) කළමනාකරණය සරල කරයි.

Digital circuits use logic gates with binary inputs to perform basic operations (AND, OR, NOT), forming the foundation of digital computing.
සංඛ්‍යාංක පරිගනකයේ පදනම සාදමින් මූලික මෙහෙයුම් (AND, OR, NOT) සිදු කිරීම සඳහා සංඛ්‍යාංක පරිපථ ද්විමය යෙදවුම් සහිත තාර්කික ද්වාර භාවිතා කරයි.

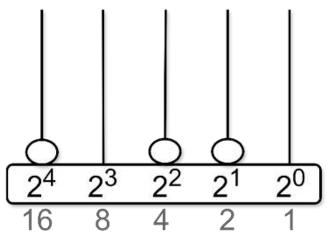
Binary numbers align with Boolean algebra, supporting digital circuit design and analysis.
ද්විමය සංඛ්‍යා බූලියන් වීජ ගණිතය සමඟ සමපාත වන අතර, සංඛ්‍යාංක පරිපථ නිර්මාණයට සහ විශ්ලේෂණයට සහාය වේ.

Binary arithmetic is simpler than decimal in some ways; addition only requires carrying when adding two 1s, making hardware implementation easier.
ද්විමය ගණිතය සමහර පැතිවලින් දශම ගණිතයට වඩා සරල ය. උදාහරණයක් ලෙස, එකතු කිරීම අවශ්‍ය වන්නේ 1 දෙකක් සාරාංශ කරන විට පමණක් රැගෙන යාමයි, එමඟින් දෘඩාංගයේ ක්‍රියාත්මක කිරීම සරල කරයි.

Binary efficiently represents integers and fractions using fixed-point and floating-point formats.
ද්විමය කාර්යක්ෂමව ස්ථාවර ලක්ෂ්‍ය සහ ඉපිලෙන ලක්ෂ්‍ය ආකෘති භාවිතා කරමින් පූර්ණ සංඛ්‍යා සහ භාග නියෝජනය කරයි.

Many data communication protocols and file formats rely on binary encoding, ensuring that different devices and systems can communicate effectively.
බොහෝ දත්ත සන්නිවේදන නියමාවලි සහ ගොනු ආකෘති මගින් විවිධ උපාංග සහ පද්ධති වලට එලදායී ලෙස සන්නිවේදනය කළ හැකි බව සහතික කරමින් ද්විමය කේතනය මත රඳා පවතී.

Let's consider a binary number 10110_2.
10110_2 ද්විමය අංකය සලකමු.



Addition of two base 2 numbers
2 පාදයේ සංඛ්‍යා දෙකක් එකතු කිරීම

Example 01:
$= 10111_2 + 1111_2$

32 2^5	16 2^4	8 2^3	4 2^2	2 2^1	1 2^0
1	1	1	1	1	
0	1	0	1	1	1_2
	+	1	1	1	1_2
1	2	2	3	3	2
	-2	-2	-2	-2	-2
1	0	0	1	1	0_2

Example 02:

= $101011_2 + 11111_2$

64	32	16	8	4	2	1
2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	1	1	1	
	1	0	1	0	1	1_2
	+	1	1	1	1	1_2
1	2	2	3	2	3	2
	-2	-2	-2	-2	-2	-2
1	0	0	1	0	1	0_2

$11010110_2 + 01101101_2 = ?$

Subtraction of two base 2 numbers

2 පාදයේ සංඛ්‍යා දෙකක් අඩු කිරීම

Example 01:

= $101011_2 - 1111_2$

32	16	8	4	2	1
2^5	2^4	2^3	2^2	2^1	2^0
	~~1~~ ~~2~~	2	2		
0 ~~1~~	0	0 ~~1~~	0	1	1_2
-	0	1	1	1	1_2
0	1	1	1	0	0_2

Example 02:

= $10111_2 - 1111_2$

16	8	4	2	1
2^4	2^3	2^2	2^1	2^0
	2			
0 ~~1~~	0	1	1	1_2
-	1	1	1	1_2
	1	0	0	0_2

3. Octal Number System

The octal number system is a base-8 system, which means it uses eight digits 0, 1, 2, 3, 4, 5, 6, and 7.

අෂ්ඨමය සංඛ්‍යා පද්ධතිය යනු පාද 8 පද්ධතියකි, එයින් අදහස් වන්නේ එය ඉලක්කම් අටක් 0, 1, 2, 3, 4, 5, 6 සහ 7 භාවිතා කරන බවයි.

In the octal system, the place value of each digit depends on its position, starting from the right. The place values are powers of 8.

අෂ්ඨමය ක්‍රමයේදී, එක් එක් ඉලක්කම්වල ස්ථාන අගය දකුණේ සිට ආරම්භ වන එහි පිහිටීම මත රඳා පවතී. ස්ථාන අගයන් 8 බල වේ.

8^4	8^3	8^2	8^1	8^0
4096	512	64	8	1

Octal numbers consist only of the digits 0 to 7. No digit in an octal number can be greater than 7.

අෂ්ඨමය සංඛ්‍යා සමන්විත වන්නේ 0 සිට 7 දක්වා වූ ඉලක්කම් වලින් පමණි. අෂ්ඨමය සංඛ්‍යාවක කිසිදු ඉලක්කමක් 7 ට වඩා වැඩි නොවිය හැක.

Why are octal numbers used?

අෂ්ඨමය සංඛ්‍යා භාවිතා කරන්නේ ඇයි?

Octal numbers use only eight digits (0-7), making it easier to handle than larger bases.

අෂ්ඨමය සංඛ්‍යා ඉලක්කම් අටක් (0-7) පමණක් භාවිතා කරන අතර, එය විශාල පාදවලට වඩා හැසිරවීම පහසු වේ.

Each octal digit represents three binary digits, reducing the overall length of binary sequences.

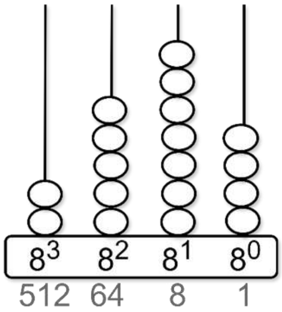
සෑම අෂ්ඨමය සංඛ්‍යාවක්ම ද්විමය ඉලක්කම් තුනක් මගින් නියෝජනය කරයි, ද්විමය අනුක්‍රමයෙහි සමස්ත දිග අවම කරයි.

Certain mathematical operations can be performed more easily in octal, especially when dealing with powers of two.

විශේෂයෙන් දෙකෙහි බලයන් සමඟ කටයුතු කරන විට සමහර ගණිතමය මෙහෙයුම් අෂ්ඨමයේ දී වඩාත් පහසුවෙන් සිදු කළ හැකිය.

Binary-to-octal conversion is easy, as each three-bit group maps directly to one octal digit.
සෑම බිටු තුනක කාණ්ඩයමක්ම සෘජුවම එක් අෂ්ඨමය ඉලක්කමක් වෙත සිතියම් ගත වන බැවින්, ද්විමය-අෂ්ඨමය පරිවර්තනය පහසු වේ.

Let's consider the octal number 2574_8.
2574_8 යන අෂ්ඨමය අංකය සලකමු.



Addition of two base 8 numbers
8 පාදයේ සංඛ්‍යා දෙකක් එකතු කිරීම

Example 01:
= $1765_8 + 732_8$

512	64	8	1
8^3	8^2	8^1	8^0
1	1		
1	7	6	5_8
+	7	3	2_8
2	15	9	7
	-8	-8	
2	7	1	7_8

Substraction of two base 8 numbers
8 පාදයේ සංඛ්‍යා දෙකක් අඩු කිරීම

Example 01:
= $1324_8 - 762_8$

512	64	8	1
8^3	8^2	8^1	8^0
	8	8	
0 ~~1~~	2 ~~3~~	2	4_8
-	7	6	2_8
	3	4	2_8

4. Hexadecimal Number System
ෂඩ්දශමය සංඛ්‍යා පද්ධතිය

The positional numeral system that uses 16 symbols to represent numbers is called hexadecimal number system (also called base 16).
සංඛ්‍යා නිරූපණය කිරීම සඳහා සංකේත 16ක් භාවිතා කරන ස්ථානීය සංඛ්‍යා පද්ධතියක් ලෙස ෂඩ්දශමය සංඛ්‍යා පද්ධතිය (16 පාදය ලෙසද හැඳින්වේ) හඳුන්වයි

Decimal	Hexa -Decimal	Tally Marks
0	0	
:	:	:
9	9	𝍸 ||||
10	A	𝍸 𝍸
11	B	𝍸 𝍸 |
12	C	𝍸 𝍸 ||
13	D	𝍸 𝍸 |||
14	E	𝍸 𝍸 ||||
15	F	𝍸 𝍸 𝍸

Hexadecimal is often used in computing as a more concise representation of binary numbers, as each hexadecimal digit represents 4 bits.
සෑම ෂඩ්දශම ඉලක්කමක්ම බිටු 4ක් නියෝජනය කරන බැවින්, ද්විමය සංඛ්‍යා වඩාත් සංක්ෂිප්ත නිරූපණයක් ලෙස පරිගණකකරණයේදී ෂඩ්දශම බොහෝ විට භාවිතා වේ.

Each position in a hexadecimal number has a place value that is a power of 16.
ෂඩ්දශම සංඛ්‍යාවක සෑම ස්ථානයකම ස්ථානීය අගයක් ඇති අතර එය 16යේ බලයකි.

For example, a 16-bit binary number like 1101101010110101 is more easily represented as DAB5 in hexadecimal.
උදාහරණයක් ලෙස, 1101101010110101 වැනි (16-bit) ද්විමය අංකයක් ෂඩ්දශමයෙන් DAB5 ලෙස වඩාත් පහසුවෙන් නිරූපණය වේ.

Why are Hexa-decimal numbers used?
ෂඩ්දශමය සංඛ්‍යා භාවිතා කරන්නේ ඇයි?

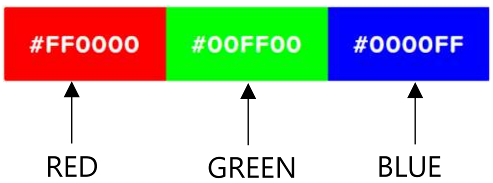
Hexa-decimal is shorter than binary. Each hexa-decimal digit shows 4 binary bits, so long binary numbers look shorter in hexa-decimal.
ෂඩ්දශම ද්විමය වලට වඩා කෙටි වේ. සෑම ෂඩ් දශම සංඛ්‍යාවක්ම ද්විමය බිටු 4ක් පෙන්වයි, එබැවින් දීර්ඝ ද්විමය සංඛ්‍යා ෂඩ් දශමය ආකාරයෙන් කෙටියෙන් නිරූපණය කළ හැක.

Hexa-decimal is used for memory addresses in computers. It makes addresses simpler to read than binary, while still fitting the way computers work.
පරිගණකවල මතක ලිපින සඳහා ෂඩ්දශම භාවිතා වේ. එය පරිගණක ක්‍රියා කරන ආකාරයට සරිලන අතරම ලිපින ද්විමය කියවීමට වඩා සරල කරයි.

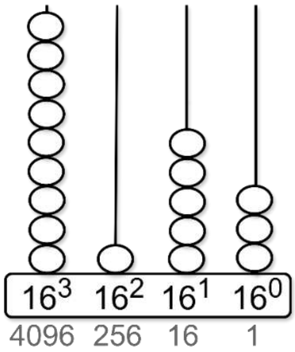
Hexa-decimal numbers make large binary values easier to understand. Programmers can use shorter hexadecimal numbers instead of long binary sequences.
ෂඩ්දශම සංඛ්‍යා විශාල ද්විමය අගයන් තේරුම් ගැනීමට පහසු කරයි. ක්‍රමලේඛකයින්ට දීර්ඝ ද්විමය අනුක්‍රමයන් වෙනුවට කෙටි ෂඩ් දශම සංඛ්‍යා භාවිතා කළ හැක.

Hexa-decimal also helps represent machine instructions and data in assembly code.
ෂඩ්දශම ඇසෙම්බ්ලි කේතයේ යන්ත්‍ර උපදෙස් සහ දත්ත නිරූපණය කිරීමට ද උපකාරී වේ.

Hexa-decimal is used in web design to show colours. For example, #FF5733 represents a mix of red, green, and blue, making it a standard way to display colours online.
වෙබ් නිර්මාණයේදී වර්ණ පෙන්වීමට ෂඩ්දශම භාවිතා වේ. උදාහරණයක් ලෙස, #FF5733 රතු, කොළ සහ නිල් මිශ්‍රණයක් නියෝජනය කරයි, එය අන්තර්ජාලය හරහා වර්ණ සංදර්ශන කිරීමට සම්මත ක්‍රමයක් බවට පත් කරයි.



Let's consider the Hexa-decimal number 9153_{16}.
9153_{16} යන ෂඩ්දශමය සංඛ්‍යාව සලකමු.



Addition of two base 16 numbers
16 පාදයේ සංඛ්‍යා දෙකක් එකතු කිරීම

Example 01:
= $ABCD_{16} + 9FE_{16}$

4096	256	16	1
16^3	16^2	16^1	16^0
1	1	1	
A	B	C	D
+	9	F	E
11	21	28	27
	-16	-16	-16
11	5	12	11
B	5	C	B_{16}

Substraction of two base 16 numbers
16 පාදයේ සංඛ්‍යා දෙකක් අඩු කිරීම

Example 01:
= $ABCD_{16} - 9FE_{16}$

4096	256	16	1
16^3	16^2	16^1	16^0
		16	16
A	10 ~~B~~	11 ~~C~~	D_{16}
-	9	F	E_{16}
10	1	12	15
A	1	C	F_{16}

Why do we convert numbers from one number system to another?
එක් සංඛ්‍යා පද්ධතියකින් තවත් සංඛ්‍යා පද්ධතියකට සංඛ්‍යා පරිවර්තනය කරන්නේ ඇයි?

1. Compatibility with Systems or Devices
පද්ධති හෝ උපාංග සමඟ ගැළපීම

Different systems or devices use different number systems. For example, computers use the binary number system, while humans typically use the decimal system. Converting numbers allows us to interact with computers effectively.
විවිධ පද්ධති හෝ උපාංග විවිධ සංඛ්‍යා පද්ධති භාවිතා කරයි. උදාහරණයක් ලෙස, පරිගණක ද්විමය සංඛ්‍යා පද්ධතිය භාවිතා කරන අතර මිනිසුන් සාමාන්‍යයෙන් දශමය පද්ධතිය භාවිතා කරයි. සංඛ්‍යා පරිවර්තනය කිරීමෙන් අපට පරිගණක සමඟ ඵලදායී ලෙස අන්තර් ක්‍රියා කිරීමට ඉඩ සලසයි.

2. Ease of Calculation
ගණනය කිරීමේ පහසුව

Some number systems, like binary or hexadecimal, are more suited to specific tasks or calculations.
ද්විමය හෝ ෂඩ්දශම වැනි සමහර සංඛ්‍යා පද්ධති විශේෂිත කාර්යයන් හෝ ගණනය කිරීම් සඳහා වඩාත් ගැලපේ.

3. Efficiency in Storage and Transmission
ගබඩා කිරීමේ සහ සම්ප්‍රේෂණයේ කාර්යක්ෂමතාව

Some number systems, like binary, are more efficient for storing and transmitting data in digital systems. Data is often converted to binary for better performance in digital devices.
ද්විමය වැනි සමහර සංඛ්‍යා පද්ධති අංකිත පද්ධතිවල දත්ත ගබඩා කිරීම සහ සම්ප්‍රේෂණය කිරීම සඳහා වඩාත් කාර්යක්ෂම වේ. අංකිත උපාංගවල වඩා හොඳ කාර්ය සාධනය සඳහා දත්ත බොහෝ විට ද්විමය බවට පරිවර්තනය වේ.

4. Data Encryption and Security
දත්ත සංකේතනය සහ ආරක්ෂාව

Number conversions are often part of encryption processes to protect sensitive data by converting it into forms that are difficult to understand without the correct key.
නිවැරදි යතුර නොමැතිව තේරුම් ගැනීමට අපහසු ආකෘති බවට පරිවර්තනය කිරීමෙන් සංවේදී දත්ත ආරක්ෂා කිරීම සඳහා සංඛ්‍යා පරිවර්තනය බොහෝ විට සංකේතන ක්‍රියාවලීන්ගේ කොටසකි.

5. Scientific and Engineering Calculations
විද්‍යාත්මක සහ ඉංජිනේරු ගණනය කිරීම්

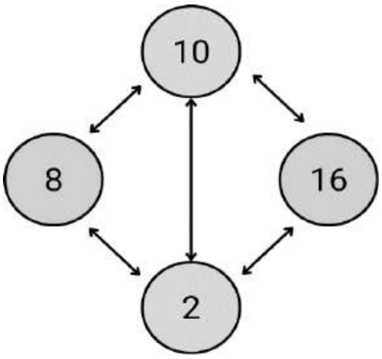
Some fields, like engineering or physics, use number systems that simplify complex calculations, such as using scientific notation or hexadecimal for very large or very small numbers.
ඉංජිනේරු හෝ භෞතික විද්‍යාව වැනි සමහර ක්ෂේත්‍ර, ඉතා විශාල හෝ ඉතා කුඩා සංඛ්‍යා සඳහා විද්‍යාත්මක අංකනය හෝ ෂඩ් දශම භාවිතා කිරීම වැනි සංකීර්ණ ගණනය කිරීම් සරල කරන සංඛ්‍යා පද්ධති භාවිතා කරයි.

6. Human Readability
මානව කියවීමේ හැකියාව

Humans are more familiar with the decimal system, so converting numbers from other systems (like binary or hexadecimal) makes it easier for people to read and interpret.
මිනිසුන් දශමය ක්‍රමයට වඩා හුරුපුරුදුය, එබැවින් වෙනත් පද්ධති වලින් සංඛ්‍යා පරිවර්තනය කිරීම (ද්විමය හෝ ෂඩ් දශම වැනි) මිනිසුන්ට කියවීමට සහ අර්ථ නිරූපණය කිරීමට පහසු කරයි.

Converting Between Number Bases
සංඛ්‍යා පාද අතර පරිවර්තනය

Converting between number bases is the process of transforming a number from one numeral system (or base) to another, while maintaining its original value.
සංඛ්‍යා පාද අතර පරිවර්තනය යනු සංඛ්‍යාවක මුල් අගය පවත්වා ගනිමින් එය එක් සංඛ්‍යා පද්ධතියකින් (හෝ පාදයේ) තවත් සංඛ්‍යාවකට පරිවර්තනය කිරීමේ ක්‍රියාවලියයි.



This is essential when working with different systems, such as converting between binary (base-2), decimal (base-10), octal (base-8), or hexa-decimal (base-16).
ද්වීමය, දශමය, අෂ්ඨමය හෝ ෂඩ්දශමය අතර පරිවර්තනය කිරීම වැනි විවිධ පද්ධති සමඟ වැඩ කිරීමේදී මෙය අත්‍යවශ්‍ය වේ.

It allows for easier interpretation of numbers depending on the context binary for computers, decimal for everyday use, and hexa-decimal for programming or digital electronics.
පරිගණක සඳහා ද්වීමය, එදිනෙදා භාවිතය සඳහා දශමය සහ ක්‍රමලේඛනය හෝ අංකිත ඉලෙක්ට්‍රොනික සඳහා ෂඩ්දශම යන සන්දර්භය මත පදනම්ව සංඛ්‍යා පහසුවෙන් අර්ථ නිරූපණය කිරීමට එය ඉඩ සලසයි.

A. Converting from Decimal to Binary
දශමය සංඛ්‍යා ද්වීමය බවට හැරවීම

Method 01

Divide the given base 10 number from 2 one step at a time continuously until it finally becomes zero.
දී ඇති දහයේ පාදයේ සංඛ්‍යාව අවසානයේ ශූන්‍ය වන තෙක් පියවරෙන් පියවර දිගින් දිගටම නොනවත්වා 2 න් බෙදන්න

Write the remainder in front of each step.
සෑම පියවරකටම ඉදිරියෙන් ඉතිරි අගය ලියන්න

Each remainder can be either 0 or 1.
සෑම ඉතිරි අගයක්ම 0 හෝ 1 විය හැක.

Arrange the remainders from bottom to top. It will give the equal base 2 number.
අග සිට මුලට පිහිටන පරිදි එම ඉතිරි අගයන් පෙලගස්වන්න. එමගින් අදාල සංඛ්‍යාව ලැබේ.

Example 01

$= 27_{10} \rightarrow (\quad)_2$

2	27	
2	13	- 1
2	6	- 1
2	3	- 0
2	1	- 1
	0	- 1

$27_{10} = 11011_2$

Example 02

$= 111_{10} \rightarrow (\quad)_2$

2	111	
2	55	- 1
2	27	- 1
2	13	- 1
2	6	- 1
2	3	- 0
2	1	- 1
	0	- 1

$111_{10} = 1101111_2$

Method 02

Find the largest power of 2 that is less than or equal to the decimal number.
දශමය සංඛ්‍යාවට වඩා අඩු හෝ සමාන වන 2 හි විශාලතම බලය සොයා ගන්න.

Subtract the power of 2 from the decimal number.
දශමය සංඛ්‍යාවෙන් 2 හි බලය අඩු කරන්න.

Note the 1 in binary position for that 2 power.
එම 2 බලය සඳහා ද්විමය පිහිටුමේ 1 සටහන් කරන්න.

Continue reducing the largest power of 2 until the remainder is zero, and repeat the process with the remainder.
ඉතිරිය ශුන්‍ය වන තෙක් 2 හි විශාලතම බල අඩු කිරීම දිගටම කරගෙන, ඉතිරිය සමඟ ක්‍රියාවලිය නැවත නැවත සිදු කරන්න.

For any power of 2 that is irreducible (that is, it was larger), record a 0 in that binary position.
අඩු කළ නොහැකි 2 හි ඕනෑම බලයක් සඳහා (එනම්, එය විශාල විය), එම ද්විමය ස්ථානයේ 0 සටහන් කරන්න.

Example 01

= $27_{10} \rightarrow (\quad)_2$

16	8	4	2	1
2^4	2^3	2^2	2^1	2^0
1	1	0	1	1
11011_2				

$27_{10} = 11011_2$

$$\begin{array}{rr} & 27 \\ - & 16 \\ \hline & 11 \\ - & 8 \\ \hline & 3 \\ - & 2 \\ \hline & 1 \\ - & 1 \\ \hline & 0 \\ \hline\hline \end{array}$$

Example 02

= $55_{10} \rightarrow (\quad)_2$

32	16	8	4	2	1
2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	1	1	1

$55_{10} = 110111_2$

$$\begin{array}{rr} & 55 \\ - & 32 \\ \hline & 23 \\ - & 16 \\ \hline & 7 \\ - & 4 \\ \hline & 3 \\ - & 2 \\ \hline & 1 \\ - & 1 \\ \hline & 0 \\ \hline\hline \end{array}$$

Example 03

= $78_{10} \rightarrow (\quad)_2$

64	32	16	8	4	2	1
2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0	1	1	1	0

$68_{10} = 1001110_2$

$$\begin{array}{rr} & 78 \\ - & 64 \\ \hline & 14 \\ - & 8 \\ \hline & 6 \\ - & 4 \\ \hline & 2 \\ - & 2 \\ \hline & 0 \\ \hline\hline \end{array}$$

Example 04

= $45_{10} \rightarrow (\quad)_2$

32	16	8	4	2	1
2^5	2^4	2^3	2^2	2^1	2^0
1	0	1	1	0	1

$45_{10} = 101101_2$

$$\begin{array}{rr} & 45 \\ - & 32 \\ \hline & 13 \\ - & 8 \\ \hline & 5 \\ - & 4 \\ \hline & 1 \\ \hline\hline \end{array}$$

B. Converting from Decimal to Octal
දශමය සංඛ්‍යා අෂ්ඨමය බවට හැරවීම

Divide the given base 10 number from 8 one step at a time continuously until it finally becomes zero.
දී ඇති දහයේ පාදයේ සංඛ්‍යාව අවසානයේ ශූන්‍ය වන තෙක් පියවරෙන් පියවර දිගින් දිගටම නොනවත්වා 8 න් බෙදන්න

Write the remainder in front of each step.
සෑම පියවරකටම ඉදිරියෙන් ඉතිරි අගය ලියන්න

Each remainder can be 0,1,2,3,4,5,6,7
සෑම ඉතිරි අගයක්ම 0,1,2,3,4,5,6,7 විය හැක.

Arrange the remainders from bottom to top. It will give the equal base 8 number.
අග සිට මුලට පිහිටන පරිදි එම ඉතිරි අගයන් පෙලගස්වන්න. එමගින් අදාල සංඛ්‍යාව ලැබේ.

Example 01

= $27_{10} \rightarrow (\quad)_8$

8	27	
8	3	- 3
	0	- 3

$27_{10} = 33_8$

Example 02

= $111_{10} \rightarrow (\quad)_8$

8	111	
8	13	- 7
8	1	- 5
	0	- 1

$111_{10} = 157_8$

Example 03

= $56_{10} \rightarrow (\quad)_8$

8	56	
8	7	- 0
	0	- 7

$56_{10} = 70_8$

Example 04

= $286_{10} \rightarrow (\quad)_8$

8	286	
8	35	- 6
8	4	- 3
	0	- 4

$286_{10} = 436_8$

Example 05

= $19_{10} \rightarrow (\quad)_8$

8	19	
8	2	- 3
	0	- 2

$19_{10} = 23_8$

Example 06

= $792_{10} \rightarrow (\quad)_8$

8	792	
8	99	- 0
8	12	- 3
8	1	- 4
	0	- 1

$792_{10} = 1430_8$

C. Converting from Decimal to Hexa-decimal

දශමය සංඛ්‍යා ෂඩ්දශමය බවට හැරවීම

Divide the given base 10 number by 16 one step at a time continuously until it finally becomes zero.
දී ඇති දහයේ පාදයේ සංඛ්‍යාව අවසානයේ ශුන්‍ය වන තෙක් පියවරෙන් පියවර දිගින් දිගටම නොනවත්වා 16 න් බෙදන්න

Write the remainder in front of each step.
සෑම පියවරකටම ඉදිරියෙන් ඉතිරි අගය ලියන්න

Each remainder can be as follows.
සෑම ඉතිරි අගයක්ම ඉහත පරිදි විය හැක.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Arrange the remainders from bottom to top. It will give an equal base 16 number.
අග සිට මුලට පිහිටන පරිදි එම ඉතිරි අගයන් පෙලගස්වන්න. එමගින් අදාළ සංඛ්‍යාව ලැබේ.

Example 01

$= 27_{10} \rightarrow (\quad)_{16}$

16	27	
16	1	- 11 (B)
	0	- 1

$27_{10} = 1B_{16}$

Example 02

$= 111_{10} \rightarrow (\quad)_{16}$

16	111	
16	6	- 15 (F)
	0	- 6

$111_{10} = 6F_{16}$

Example 03

$= 65_{10} \rightarrow (\quad)_{16}$

16	65	
16	4	- 1
	0	- 4

$65_{10} = 41_{16}$

Example 04

$= 283_{10} \rightarrow (\quad)_{16}$

16	283	
16	17	- 11 (B)
16	1	- 1
	0	- 1

$283_{10} = 11B_{16}$

NOTE:

Converting from Decimal to Base N

දශමය සංඛ්‍යා N පාදය බවට හැරවීම

Divide the given base 10 number from n one step at a time continuously until it finally becomes zero.
දී ඇති දහයේ පාදයේ සංඛ්‍යාව අවසානයේ ශුන්‍ය වන තෙක් පියවරෙන් පියවර දිගින් දිගටම නොනවත්වා n ගෙන් බෙදන්න

Write the remainder in front of each step.
සෑම පියවරකටම ඉදිරියෙන් ඉතිරි අගය ලියන්න

Each remainder can be 0 upto n-1.
සෑම ඉතිරි අගයක්ම 0 සිට n-1 දක්වා විය හැක.

Arrange the remainders from bottom to top. It will give the equal base n number.
අග සිට මුලට පිහිටන පරිදි එම ඉතිරි අගයන් පෙලගස්වන්න. එමගින් අදාල සංඛ්‍යාව ලැබේ.

D. Converting Decimal floating-point numbers to Binary floating-point.

දශමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා ද්වීමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා බවට හැරවීම

Multiply the given decimal floating-point number by 2 one time.
දී ඇති ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා අගය එක් වරක් 2 න් ගුණ කරන්න

Neglect the integer part from the above calculation.
ඉහත ගණනය කිරීමෙන් ලැබෙන පූර්ණ සංඛ්‍යා කොටස නොසලකා හරින්න.

Take the floating-point number part only and continue multiplying from 2 as above again and again until it becomes zero.
ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා කොටස පමණක් ගෙන ඉහත පරිදි එය 0 වන තුරුම නැවත නැවත 2 න් ගුණ කරන්න

Arrange the integer values in front of the floating-point parts from top to bottom.
It will give the relevant number.
ඉහළ සිට පහළට පිහිටන සේ ඉපිලුම් ලක්ෂ්‍ය කොටස් ඉදිරිපිට ඇති පූර්ණ සංඛ්‍යා අගයන් සකසන්න. එමගින් අදාළ සංඛ්‍යාව ලබා දෙනු ඇත.

Example 01

$= 0.3125_{10} \rightarrow (\quad)_2$

	.3125	x 2
0	.6250	x 2
1	.2500	x 2
0	.5000	x 2
1	.0000	

$0.3125_{10} = 0.0101_2$

Example 02

$= 0.125_{10} \rightarrow (\quad)_2$

	.125	x 2
0	.250	x 2
0	.500	x 2
1	.000	

$0.125_{10} = 0.001_2$

NOTE:

Some numbers don’t produce zero in multiplying. If the same calculation is recurring, stop the multiplication.
ඇතැම් සංඛ්‍යා ගුණකිරීමෙන් ශුන්‍ය ලබාදෙන්නේ නැත. එකම ගණිත කර්ම නැවත නැවත සිදු වන්නේ නම්, ගුණ කිරීම නවත්වන්න.

Example 01

$= 0.2_{10} \rightarrow (\quad)_2$

	.2	x 2
0	.4	x 2
0	.8	x 2
1	.6	x 2
1.	.2	x 2
0	.4	x 2
0	.8	x 2
1	.6	

$0.2_{10} = 0.0011_2$

Example 02

$= 0.45_{10} \rightarrow (\quad)_2$

	.45	x 2
0	.90	x 2
1	.80	x 2
1	.60	x 2
1.	.20	x 2
0	.40	x 2
0	.80	x 2
1	.60	x 2
1	.20	x 2
0	.40	

$0.45_{10} = 0.01110_2$

E. Converting Decimal floating-point numbers to Octal floating-point.

දශමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා අෂ්ඨමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා බවට හැරවීම

Multiply the given decimal floating-point number by 8 one time.
දී ඇති ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා අගය එක් වරක් 8 න් ගුණ කරන්න

Neglect the integer part from the above calculation.
ඉහත ගණනය කිරීමෙන් ලැබෙන පූර්ණ සංඛ්‍යා කොටස නොසලකා හරින්න

Take the floating-point number part only and continue multiplying from 8 as above again and again until it becomes zero.
ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා කොටස පමණක් ගෙන ඉහත පරිදි එය 0 වන තුරුම නැවත නැවත 8 න් ගුණ කරන්න

Arrange the integer values in front of the floating-point parts from top to bottom.
It will give the relevant number.
ඉහළ සිට පහළට පිහිටන සේ ඉපිලුම් ලක්ෂ්‍ය කොටස් ඉදිරිපිට ඇති පූර්ණ සංඛ්‍යා අගයන් සකසන්න. එමගින් අදාළ සංඛ්‍යාව ලබා දෙනු ඇත.

Example 01

$= 0.3125_{10} \rightarrow (\quad)_8$

	.3125	x 8
2	.5000	x 8
4	.0000	

$0.3125_{10} = 0.24_8$

Example 02

$= 0.625_{10} \rightarrow (\quad)_8$

	.625	x 8
5	.000	

$0.625_{10} = 0.5_8$

NOTE:
Some numbers don't produce zero in multiplying. If the same calculation is recurring, stop the multiplication.
ඇතැම් සංඛ්‍යා ගුණකිරීමෙන් ශූන්‍ය ලබාදෙන්නේ නැත. එකම ගණිත කර්ම නැවත නැවත සිදු වන්නේ නම්, ගුණ කිරීම නවත්වන්න.

Example 01

$= 0.2_{10} \rightarrow (\quad)_8$

	.2	x 8
1	.6	x 8
4	.8	x 8
6	.4	x 8
3	.2	x 8
1	.6	x 8
4	.8	

$0.2_{10} = 0.1463_8$

Example 02

$= 0.45_{10} \rightarrow (\quad)_8$

	.45	x 8
3	.60	x 8
4	.80	x 8
6	.40	x 8
3	.20	x 8
1	.60	x 8
4	.80	x 8
6	.40	

$0.45_{10} = 0.34631_8$

F. Converting Decimal floating-point numbers to Hexa floating-point.

දශමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා ශඩ්දශමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා බවට හැරවීම

Multiply the given decimal floating-point number by 16 one time.
දී ඇති ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා අගය එක් වරක් 16 න් ගුණ කරන්න

Neglect the integer part from the above calculation.
ඉහත ගණනය කිරීමෙන් ලැබෙන පූර්ණ සංඛ්‍යා කොටස නොසලකාහරින්න

Take the floating-point number part only and continue multiplying from 16 as above again & again until it becomes zero.
ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා කොටස පමණක් ගෙන ඉහත පරිදි එය 0 වන තුරුම නැවත නැවත 16 න් ගුණ කරන්න

Arrange the integer values in front of the floating-point parts from top to bottom.
It will give the relevant number.
ඉහළ සිට පහළට පිහිටන සේ ඉපිලුම් ලක්ෂ්‍ය කොටස් ඉදිරිපිට ඇති පූර්ණ සංඛ්‍යා අගයන් සකසන්න. එමගින් අදාළ සංඛ්‍යාව ලබා දෙනු ඇත.

Example 01

$= 0.3125_{10} \rightarrow (\quad)_{16}$

	.3125	x 16
5	.0000	

$0.3125_{10} = 0.5_{16}$

Example 02

$= 0.625_{10} \rightarrow (\quad)_{16}$

	.625	x 16
10	.000	

$0.625_{10} = 0.A_{16}$

Example 03

$= 0.125_{10} \rightarrow (\quad)_{16}$

	.125	x 16
2	.000	

$0.125_{10} = 0.2_{16}$

NOTE: Some numbers don't produce zero in multiplying. If the same calculation is recurring, stop the multiplication.
ඇතැම් සංඛ්‍යා ගුණකිරීමෙන් ශුන්‍ය ලබාදෙන්නේ නැත. එකම ගණිත කර්ම නැවත නැවත සිදු වන්නේ නම්, ගුණ කිරීම නවත්වන්න.

Example 01

$= 0.2_{10} \rightarrow (\quad)_{16}$

	.2	x 16
3	.2	x 16
3	.2	x 16

$0.2_{10} = 0.3_{16}$

Example 02

$= 0.45_{10} \rightarrow (\quad)_{16}$

	.45	x 16
7	.20	x 16
3	.20	x 16
3	.20	

$0.45_{10} = 0.73_{16}$

NOTE:

Converting Decimal floating-point numbers to Base N floating-point.

දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා N පාදයේ ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට හැරවීම

Multiply the given decimal floating-point number by n one time.

දී ඇති ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා අගය එක් වරක් n ගෙන් ගුණ කරන්න

Neglect the integer part from the above calculation.

ඉහත ගණනය කිරීමෙන් ලැබෙන පූර්ණ සංඛ්‍යා කොටස නොසලකා හරින්න

Take the floating-point number part only and continue multiplying from n as above again and again until it becomes zero.

ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා කොටස පමණක් ගෙන ඉහත පරිදි එය 0 වන තුරුම නැවත නැවත n ගෙන් ගුණ කරන්න

Arrange the integer values in front of the floating-point parts from top to bottom. It will give the relevant number.

ඉහළ සිට පහළට පිහිටන සේ ඉපිලුම් ලක්ෂ්‍ය කොටස් ඉදිරිපිට ඇති පූර්ණ සංඛ්‍යා අගයන් සකසන්න. එමගින් අදාළ සංඛ්‍යාව ලබා දෙනු ඇත.

G. Converting from Binary to Decimal

ද්වීමය සංඛ්‍යා දශමය බවට හැරවීම

Method 01

Write place values of each position below the given number in powers of base 2

දී ඇති සංඛ්‍යාවට පහලින් එක් එක් ස්ථාන වල ස්ථානීය අගයන් 2 පාදයේ බල ලෙස ලියාගන්න

Convert those place values into Decimal and write them down below each position.

එම ස්ථානීය අගයන් දශමය වටිනාකම් බවට පරිවර්තනය කර ස්ථානයට පහලින් ලියන්න

Multiply digits of the binary number from the relevant decimal place values

ද්වීමය සංඛ්‍යාවේ ඉලක්කම් අදාල ස්ථාන වල දශමය ස්ථානීය වටිනාකම් වලින් ගුණ කරන්න

Add all products to get the equivalent decimal number.

අදාල දශමය සංඛ්‍යාව ලබාගැනීම සඳහා ගුණිතයන් සියල්ල එකතු කරන්න.

Example 01

= $11011_2 \rightarrow (\quad)_{10}$

11011_2

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1

1	1	0	1	1
16ε 8ε 2ε 1‿ 27				
27_{10}				
11011_2‿ 27_{10}				

Example 02

= $110111_2 \rightarrow (\quad)_{10}$

110111_2

2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1

1	1	0	1	1	1
32ε 16ε 4ε 2ε 1‿ 55					
55_{10}					
110111_2‿ 55_{10}					

Example 03

= $1101111_2 \rightarrow (\quad)_{10}$

1101111_2

2^6	2^5	2^4	2^3	2^2	2^1	2^0
64	32	16	8	4	2	1

1	1	0	1	1	1	1
64ε 32ε 8ε 4ε 2ε 1ᴗ 111_{10}						
$1101111_2 = 111_{10}$						

Method 02

Start with the first binary digit.
පළමු ද්විමය ඉලක්කම් සමඟ ආරම්භ කරන්න.

Then multiply it by 2 to get the value.
ඉන්පසු එය 2 න් ගුණ කර අගය ලබා ගන්න.

Then add with the next binary digit.
ඉන්පසු ඊළඟ ද්විමය ඉලක්කම සමග එකතු කරන්න.

Continue multiplying by 2 and adding binary digits in this way until all the digits are gone through.
එලෙස සියලු ඉලක්කම් හරහා යන තෙක් 2 න් ගුණ කිරීම සහ ද්වීමය ඉලක්කම් එකතු කිරීම දිගටම කරගෙන යන්න.

Then finally get the correct decimal value.
එවිට අවසානයේ නිවැරදි දශමය අගය ලැබේ.

Example 01

= $11011_2 \rightarrow (\quad)_{10}$

1 1 0 1 1

x2 → 2
3
x2 → 6
6
x2 → 12
13
x2 → 26
27

$11011_2 = 27_{10}$

Example 02

= $110111_2 \rightarrow (\quad)_{10}$

1 1 0 1 1 1

x2 → 2
3
x2 → 6
6
x2 → 12
13
x2 → 26
27
x2 → 54
55

$110111_2 = 55_{10}$

Example 03

= $100100_2 \rightarrow (\quad)_{10}$

1 0 0 1 0 0

x2 → 2
2
x2 → 4
4
x2 → 8
9
x2 → 18
18
x2 → 36
36

$110111_2 = 36_{10}$

H. Converting Binary floating-point numbers to Decimal floating-point.
ද්විමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට හැරවීම

This is an extension of the above method.
මෙය ඉහත ක්‍රමයේම දිගුවකි.

Place values will be negative powers of base 2.
ස්ථාන අගයන් 2 පාදයේ සෘණ බල ලෙස ලැබේ.

They will produce fractions as the equivalent decimal number when converted into decimal.
ඒවා ප්‍රසාරණය කල විට සමාන දශම සංඛ්‍යා ලෙස භාග සංඛ්‍යා නිපදවනු ඇත.

We should convert them into floating point numbers of decimal before multiplying with the relevant binary digit.
අදාළ ද්විමය ඉලක්කම් සමඟ ගුණ කිරීමට පෙර ඒවා දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට පරිවර්තනය කරගත යුතු

Example 01

$0.001_2 \rightarrow (\quad)_{10}$

2^0		2^{-1}	2^{-2}	2^{-3}
1		0.5	0.25	0.125

0	.	0	0	1
0	.	0+0+0.125		
0.125_{10}				
$0.001_2 = 0.125_{10}$				

Example 02

$110111.101_2 \rightarrow (\quad)_{10}$

2^5	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}
32	16	8	4	2	1		0.5	0.25	0.125

1	1	0	1	1	1	.	1	0	1
32+16+0+4+2+1						.	0.5+0+0.125		

55.625_{10}
$110111.101_2 = 55.625_{10}$

Example 03

$11011.0101_2 \rightarrow (\quad)_{10}$

2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}
16	8	4	2	1		0.5	0.25	0.125	0.0625

1	1	0	1	1	.	0	1	0	1
16+8+0+2+1					.	0+0.25+0+0.0625			

27.3125_{10}
$11011.0101_2 = 27.3125_{10}$

I. Converting Octal to Decimal
අෂ්ඨමය සංඛ්‍යා දශමය බවට හැරවීම

Method 01

Write place values of each position below the given number in powers of base 8
දී ඇති සංඛ්‍යාවට පහලින් එක් එක් ස්ථාන වල ස්ථානීය අගයන් 8 පාදයේ බල ලෙස ලියාගන්න

Convert those place values into Decimal and write them down below each position.
එම ස්ථානීය අගයන් දශමය වටිනාකම් බවට පරිවර්තනය කර ස්ථානයට පහලින් ලියන්න

Multiply digits of the octal number from the relevant decimal place values.
අෂ්ඨමය සංඛ්‍යාවේ ඉලක්කම් අදාල ස්ථාන වල දශමය ස්ථානීය වටිනාකම් වලින් ගුණ කරන්න

Add all products to get the equivalent decimal number.
අදාල දශමය සංඛ්‍යාව ලබාගැනීම සඳහා ගුණිතයන් සියල්ල එකතු කරන්න.

Example 01

$33_8 \rightarrow (\quad)_{10}$

8^1	8^0
8	1

3	3
3x8	3x1
24	3

27_{10}
$33_8 = 27_{10}$

Example 02

$157_8 \rightarrow (\quad)_{10}$

8^2	8^1	8^0
64	8	1

1	5	7
1x64	5x8	7x1
$64 + 40 + 7 = 111_{10}$		
$157_8 = 111_{10}$		

Method 02

Start with the first Octal digit.
පළමු අශ්ඨමය ඉලක්කම් සමඟ ආරම්භ කරන්න.

Then multiply it by 8 to get the value.
ඉන්පසු එය 8 න් ගුණ කර අගය ලබා ගන්න.

Then add with the next Octal digit.
ඉන්පසු ඊළඟ අශ්ඨමය ඉලක්කම සමග එකතු කරන්න.

Continue multiplying by 8 and adding Octal digits in this way until all the digits are gone through.
එලෙස සියලු ඉලක්කම් හරහා යන තෙක් 8 න් ගුණ කිරීම සහ අශ්ඨමය ඉලක්කම් එකතු කිරීම දිගටම කරගෙන යන්න.

Then finally get the correct decimal value.
එවිට අවසානයේ නිවැරදි දශමය අගය ලැබේ.

Example 01

$33_8 \rightarrow (\quad)_{10}$

3 3
x8 24
27

$33_8 = 27_{10}$

Example 02

$157_8 \rightarrow (\quad)_{10}$

1 5 7
x8 8
13
104
111

$157_8 = 111_{10}$

J. Converting Octal floating-point numbers to Decimal floating-point.

අෂ්ඨමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට හැරවීම

This is an extension of the above method.
මෙය ඉහත ක්‍රමයේම දිගුවකි.

Place values will be negative powers of base 8.
ස්ථාන අගයන් 8 පාදයේ සෘණ බල ලෙස ලැබේ.

They will produce fractions as the equivalent decimal number when converted into decimal.
ඒවා ප්‍රසාරණය කල විට සමාන දශමය සංඛ්‍යා ලෙස භාග සංඛ්‍යා නිපදවනු ඇත.

We should convert them into floating point numbers of decimal before multiplying with the relevant octal digit.
අදාළ අෂ්ඨමය ඉලක්කම් සමඟ ගුණ කිරීමට පෙර ඒවා දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට පරිවර්තනය කරගත යුතුය.

Example 01

$67.5_8 \rightarrow (\quad)_{10}$

8^1	8^0		8^{-1}
8	1		0.125

6	7	.	5
6 x 8	7 x 1	.	5 x 0.125
48	7		0.625
55		.	0.625

55.625_{10}
$67.5_8 = 55.625_{10}$

Example 02

$33.24_8 \rightarrow (\quad)_{10}$

8^1	8^0		8^{-1}	8^{-2}
8	1		0.125	0.015625

3	3	.	2	4
3x8	3x1	.	2x0.125	4x0.015625
27			0.3125	
27.3125_{10}				
$33.24_8 = 27.3125_{10}$				

Example 03

$58.3_8 \rightarrow (\quad)_{10}$

8^1	8^0		8^{-1}
8	1		0.125

5	8	.	3
5 x 8	8 x 1	.	3 x 0.125
40	8		0.375
48		.	0.375

48.375
$58.3_8 = 48.375_{10}$

Example 04

$51.86_8 \rightarrow (\quad)_{10}$

8^1	8^0		8^{-1}	8^{-2}
8	1		0.125	0.015625

5	1	.	8	6
5x8	1x1	.	8x0.125	6x0.015625
41			1.0937	
42.0937_{10}				
$51.86_8 = 42.0937_{10}$				

Example 05

$91.6 \rightarrow (\quad)_{10}$

8^1	8^0		8^{-1}
8	1		0.125

9	1	.	6
9 x 8	1 x 1	.	6 x 0.125
72	1		0.75
73		.	0.75

73.75_{10}
$91.6_8 = 73.75_{10}$

K. Converting Hexa-decimal to Decimal

ෂඩ්දශමය සංඛ්‍යා දශමය බවට හැරවීම

Method 01

Write the place values of each position below the given number in powers of base 16
දී ඇති සංඛ්‍යාවට පහළින් එක් එක් ස්ථාන වල ස්ථානීය අගයන් 16 පාදයේ බල ලෙස ලියාගන්න

Convert those place values into Decimal and write them down below each position.
එම ස්ථානීය අගයන් දශමය වටිනාකම් බවට පරිවර්තනය කර ස්ථානයට පහළින් ලියන්න

Multiply digits of the hexadecimal number from the relevant decimal place values
ෂඩ්දශමය සංඛ්‍යාවේ ඉලක්කම් අදාල ස්ථාන වල දශමය ස්ථානීය වටිනාකම් වලින් ගුණ කරන්න

Add all products to get the equivalent decimal number.
අදාල දශමය සංඛ්‍යාව ලබාගැනීම සඳහා ගුණිතයන් සියල්ල එකතු කරන්න.

Example 01

$1B_{16} \rightarrow (\quad)_{10}$

16^1	16^0
16	1

1	B
1 x 16	B x 1
16	11
16 + 11 = 27	
$1B_{16} = 27_{10}$	

Example 02

$37_{16} \rightarrow (\quad)_{10}$

16^1	16^0
16	1

3	7
3 x 16	7 x 1
48	7
48 + 7 = 55	
$37_{16} = 55_{10}$	

Method 02

Start with the first Hexadecimal digit.
පළමු ෂඩ්දශමය ඉලක්කම් සමඟ ආරම්භ කරන්න.

Then multiply it by 16 to get the value.
ඉන්පසු එය 16 න් ගුණ කර අගය ලබා ගන්න.

Then add with the next Hexadecimal digit.
ඉන්පසු ඊළඟ ෂඩ්දශමය ඉලක්කම සමඟ එකතු කරන්න.

Continue multiplying by 16 and adding Hexadecimal digits in this way until all the digits are gone through.
එලෙස සියලු ඉලක්කම් හරහා යන තෙක් 16 න් ගුණ කිරීම සහ ෂඩ්දශමය ඉලක්කම් එකතු කිරීම දිගටම කරගෙන යන්න.

Then finally get the correct decimal value.
එවිට අවසානයේ නිවැරදි දශමය අගය ලැබේ.

Example 01

$1B_{16} \rightarrow (\quad)_{10}$

1   B
x16 → 16
27

$1B_{16} = 27_{10}$

Example 02

$6F_{16} \rightarrow (\quad)_{10}$

6   F
x16 → 96
111_{10}

$6F_{16} = 111_{10}$

L. Converting Hexa-decimal floating-point numbers to Decimal floating-point.
ශඩ්දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට හැරවීම

This is an extension of the above method.
මෙය ඉහත ක්‍රමයේම දිගුවකි.

Place values will be negative powers of base 16.
ස්ථානීය අගයන් 16 පාදයේ සෘණ බල ලෙස ලැබේ.

They will produce fractions as the equivalent decimal number when converted into decimal.
ඒවා ප්‍රසාරණය කල විට සමාන දශම සංඛ්‍යා ලෙස භාග සංඛ්‍යා නිපදවනු ඇත.

We should convert them into floating point numbers of decimal before multiplying with the relevant hexa digit.
අදාළ ශඩ්දශමය ඉලක්කම් සමඟ ගුණ කිරීමට පෙර ඒවා දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට පරිවර්තනය කරගත යුතුය.

Example 01

$0.2_{16} \rightarrow (\quad)_{10}$

16^0		16^{-1}
1		0.0625

0	.	2
0 x 1		2 x .0625
0		0.125
$0.2_{16} = 0.125_{10}$		

Example 02

$1B.5_{16} \rightarrow (\quad)_{10}$

16^1	16^0		16^{-1}
16	1		0.0625

1	B	.	5
1 x 16	11 x 1		5x0.0625
27		.	0.3125
27.3125			
$1B.5_{16} = 27.3125_{10}$			

NOTE:
Converting from Base N to Decimal
N පාදයේ සංඛ්‍යා දශමය බවට හැරවීම

Write place values of each position below the given number in powers of base n
දී ඇති සංඛ්‍යාවට පහලින් එක් එක් ස්ථාන වල ස්ථානීය අගයන් n පාදයේ බල ලෙස ලියාගන්න

Convert those place values into Decimal and write them down below each position.
එම ස්ථානීය අගයන් දශමය වටිනාකම් බවට පරිවර්තනය කර ස්ථානයට පහලින් ලියන්න

Multiply digits of the Base N number from the relevant decimal place values
N පාදයේ සංඛ්‍යාවේ ඉලක්කම් අදාල ස්ථාන වල දශමය ස්ථානීය වටිනාකම් වලින් ගුණ කරන්න

Add all products to get the equivalent decimal number.
අදාල දශමය සංඛ්‍යාව ලබාගැනීම සඳහා ගුණිතයන් සියල්ල එකතු කරන්න.

Converting Base N floating-point numbers to Decimal floating-point.
N පාදයේ ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට හැරවීම

This is an extension of the above method.
මෙය ඉහත ක්‍රමයේම දිගුවකි.

Place values will be negative powers of base N.
ස්ථාන අගයන් N පාදයේ සෘණ බල ලෙස ලැබේ.

They will produce fractions as the equivalent decimal number when converted into decimal.
ඒවා ප්‍රසාරණය කල විට සමාන දශමය සංඛ්‍යා ලෙස භාග සංඛ්‍යා නිපදවනු ඇත.

We should convert them into floating point numbers of decimal before multiplying with the relevant Base N digit.
අදාළ N පාදයේ ඉලක්කම් සමඟ ගුණ කිරීමට පෙර ඒවා දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට පරිවර්තනය කරගත යුතුය.

M. Converting Octal to Binary
අෂ්ඨමය සංඛ්‍යා ද්විමය බවට හැරවීම

Represent Each digit of the octal number system in Binary using 3-bit sets.
සෑම අෂ්ඨමය ඉලක්කමක්ම බිටු 3 ද්විමය කොටසක් ලෙස නිරූපණය කරන්න.

Octal		**Binary**
0	-	000
1	-	001
2	-	010
3	-	011
4	-	100
5	-	101
6	-	110
7	-	111

Replace each digit of the given octal number from the relevant 3-bit set.
දී ඇති අෂ්ඨමය සංඛ්‍යාවේ එක් එක් ඉලක්කම් ඒවාට අදාල බිටු 3 කොටස මගින් ආදේශ කරන්න

Combine all the 3 bit sets together. It will give the relevant Binary number.
ඒවා සියල්ල එකට ලියන්න. එමගින් අදාල ද්විමය සංඛ්‍යාව ලැබේ.

Zeros in the front, if any, can be neglected.
ඉදිරිපසින් ඇති බින්දු නොසලකා හැරිය හැක

Example 01

$374_8 \rightarrow (\quad)_2$

3			7			4		
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1	4	2	1
0	1	1	1	1	1	1	0	0

$011\ 111\ 100_2$
$374_8 = 011111100_2$

Example 02

$253_8 \rightarrow (\quad)_2$

2			5			3		
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1	4	2	1
0	1	0	1	0	1	0	1	1

$010\ 101\ 011_2$
$253_8 = 010101011_2$

Example 03

$620_8 \rightarrow (\quad)_2$

6			2			0		
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1	4	2	1
1	1	0	0	1	0	0	0	0

$110\ 010\ 000_2$
$620_8 = 110010000_2$

Example 04

$274_8 \rightarrow (\quad)_2$

2			8			4		
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1	4	2	1
0	1	0	1	1	1	1	0	0

$010\ 111\ 100_2$
$274_8 = 010111100_2$

Example 05

$753_8 \rightarrow (\quad)_2$

7			5			3		
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1	4	2	1
1	1	1	1	0	1	0	1	1

$111\ 101\ 011_2$
$753_8 = 111101011_2$

N. Converting Octal floating-point numbers to Binary floating-point.

අශ්ඨමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා ද්වීමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට හැරවීම

We can use the same method as above.
මේ සඳහා ඉහත ක්‍රමයම භාවිතා කල හැකිය.

Zeros in the back, if any, can be neglected.
පසුපසින් ඇති බින්දු නොසලකා හැරිය හැක

Example 01

$67.56_8 \rightarrow (\quad)_2$

6			7			.	5			6		
2^2	2^1	2^0	2^2	2^1	2^0		2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1		4	2	1	4	2	1
1	1	0	1	1	1	.	1	0	1	1	1	0

$110\ 111.101\ 110_2$
$67.56_8 = 110111.101110_2$

Example 02

$0.321_8 \rightarrow (\quad)_2$

0			.	3			2			1		
2^2	2^1	2^0		2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1		4	2	1	4	2	1	4	2	1
0	0	0	.	0	1	1	0	1	0	0	0	1

$000.011\ 010\ 001_2$
$0.321_8 = 0.011010001_2$

Example 03

$33.734_8 \rightarrow (\quad)_2$

3			3			.	7			3			4		
2^2	2^1	2^0	2^2	2^1	2^0		2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1		4	2	1	4	2	1	4	2	1
0	1	1	0	1	1	.	1	1	1	0	1	1	1	0	0

$011\ 011.111\ 011\ 100_2$
$33.734_8 = 11011.111011100_2$

O. Converting from Hexa-decimal to Binary
ශඩ්දශමය සංඛ්‍යා ද්වීමය බවට හැරවීම

Represent Each digit of the hexa-decimal number system in Binary using 4-bit sets.
සෑම ශඩ්දශමය ඉලක්කමක්ම බිටු 4 ද්වීමය කොටසක් ලෙස නිරූපණය කරන්න.

Hexa		Binary	Hexa		Binary
0	-	0000	8	-	1000
1	-	0001	9	-	1001
2	-	0010	A	-	1010
3	-	0011	B	-	1011
4	-	0100	C	-	1100
5	-	0101	D	-	1101
6	-	0110	E	-	1110
7	-	0111	F	-	1111

Replace each digit of the given hexa-decimal number from the relevant 4-bit set.
දී ඇති ශඩ්දශමය සංඛ්‍යාවේ එක් එක් ඉලක්කම් ඒවාට අදාල බිටු 4 කොටස මගින් ආදේශ කරන්න

Combine all the 4 bit sets together. It will give the relevant Binary number.
ඒවා සියල්ල එකට ලියන්න. එමගින් අදාල ද්වීමය සංඛ්‍යාව ලැබේ.

Zeros in the front, if any, can be neglected.
ඉදිරිපසින් ඇති බින්දු නොසලකා හැරිය හැක.

Example 01

= $2FA_{16} \rightarrow (\quad)_2$

2			
2^3	2^2	2^1	2^0
8	4	2	1
0	0	1	0

F (15)			
2^3	2^2	2^1	2^0
8	4	2	1
1	1	1	1

A (10)			
2^3	2^2	2^1	2^0
8	4	2	1
1	0	1	0

0010 1111 1010
$2FA_{16} = 001011111010_2$

Example 02

= $E93_{16} \rightarrow (\quad)_2$

E (14)			
2^3	2^2	2^1	2^0
8	4	2	1
1	1	1	0

9			
2^3	2^2	2^1	2^0
8	4	2	1
1	0	0	1

3			
2^3	2^2	2^1	2^0
8	4	2	1
0	0	1	1

1110 1001 0011
$E93_{16} = 111010010011_2$

Example 03

= $857_{16} \rightarrow (\quad)_2$

8			
2^3	2^2	2^1	2^0
8	4	2	1
1	0	0	0

5			
2^3	2^2	2^1	2^0
8	4	2	1
0	1	0	1

7			
2^3	2^2	2^1	2^0
8	4	2	1
0	1	1	1

1000 0101 0111
$857_{16} = 100001010111_2$

Example 04

= $B81_{16} \rightarrow (\quad)_2$

B (11)			
2^3	2^2	2^1	2^0
8	4	2	1
1	0	1	1

8			
2^3	2^2	2^1	2^0
8	4	2	1
1	0	0	0

1			
2^3	2^2	2^1	2^0
8	4	2	1
0	0	0	1

1011 1000 0001
$B81_{16} = 101110000001_2$

Example 05

= $168_{16} \rightarrow (\quad)_2$

1			
2^3	2^2	2^1	2^0
8	4	2	1
0	0	0	1

6			
2^3	2^2	2^1	2^0
8	4	2	1
0	1	1	0

8			
2^3	2^2	2^1	2^0
8	4	2	1
1	0	0	0

0001 0110 1000
$168_{16} = 000101101000_2$

P. Converting Hexa-decimal floating-point numbers to Binary floating-point.
ෂඩ්දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා ද්වීමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට හැරවීම

We can use the same method as above.
මේ සඳහා ඉහත ක්‍රමයම භාවිතා කල හැකිය.

Zeros in the back, if any, can be neglected.
පසුපසින් ඇති බින්දු නොසලකා හැරිය හැක

Example 01
= $0.D1_{16} \rightarrow (\quad)_2$

0				.	D (13)				1			
2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1		8	4	2	1	8	4	2	1
0	0	0	0	.	1	1	0	1	0	0	0	1

$0.1101\ 0001_2$
$0.D1_{16} = 0.11010001_2$

Example 02
= $37.C6_{16} \rightarrow (\quad)_2$

3				7				.	C (12)				6			
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1		8	4	2	1	8	4	2	1
0	0	1	1	0	1	1	1	.	1	1	0	0	0	1	1	0

z

$0011\ 0111.1100\ 0110_2$
$37.C6_{16} = 110111.11000110_2$

Example 03
= $1B.AF2_{16} \rightarrow (\quad)_2$

1				B (11)				.	A (10)				F (15)				2			
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1		8	4	2	1	8	4	2	1	8	4	2	1
0	0	0	1	1	0	1	1	.	1	0	1	0	1	1	1	1	0	0	1	0

$0001\ 1011.1010\ 1111\ 0010_2$
$1B.AF2_{16} = 11011.101011110010_2$

Q. Converting from Binary to Octal
ද්වීමය සංඛ්‍යා අෂ්ඨමය බවට හැරවීම

Split the given binary integer number part from right to left, from the floating point into sets of 3 bits.
දී ඇති ද්වීමය පූර්ණ සංඛ්‍යා කොටස දශමස්ථානයේ සිට දකුණේ සිට වමට බිටු 3 කොටස් වලට වෙන්කරන්න.

If necessary, add zeros to the front to complete the 3 bit sets as needed.
බිටු 3 කොටස් සම්පූර්ණ කිරීමට අවශ්‍ය නම් අවශ්‍ය පරිදි මුලට බින්දු යොදන්න

Replace each set of 3 bits with the relevant octal digit from the above table.
සෑම බිටු 3 කොටසක්ම එයට අදාල අෂ්ඨමය ඉලක්කම මගින් ආදේශ කරන්න.

Example 01
= $111101100_2 \rightarrow (\quad)_8$

111101100_2

1	1	1	1	0	1	1	0	0
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1	4	2	1
4+2+1=7			4+1=5			4		

754_8
$111101100_2 = 754_8$

Example 02
= $1101110101_2 \rightarrow (\quad)_8$

1101110101_2

0	0	1	1	0	1	1	1	0	1	0	1
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1	4	2	1	4	2	1
1			4+1=5			4+2=6			4+1=5		

1565_8
$1101110101_2 = 1565_8$

Example 03
= $10110111011_2 \rightarrow (\quad)_8$

10110111011_2

0	1	0	1	1	0	1	1	1	0	1	1
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1	4	2	1	4	2	1
2			6			7			3		

2673_8
$10110111011_2 = 2673_8$

Example 04
= $1000100101_2 \rightarrow (\quad)_8$

1000100101_2

0	0	1	0	0	0	1	0	0	1	0	1
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1	4	2	1	4	2	1
1			0			4			5		

1045_8
$1000100101_2 = 1045_8$

Example 05
= $1110110_2 \rightarrow (\quad)_8$

1110110_2

0	0	1	1	1	0	1	1	0
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1	4	2	1
1			6			6		

166_8
$1110110_2 = 166_8$

R. Converting Binary floating-point numbers to Octal floating-point.

ද්වීමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා අශ්ඨමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා බවට හැරවීම

Split the given binary floating-point number from left to right, from the floating point into sets of 3 bits.
දී ඇති ද්වීමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා කොටස දශමස්ථානයේ සිට වමේ සිට දකුණට බිටු 3 කොටස් වලට වෙන්කරන්න

Zeros can be added to the end to complete the sets of 3 bits as necessary.
බිටු 3 කොටස් සම්පූර්ණ කිරීමට අවශ්‍ය නම් අවශ්‍ය පරිදි අගට බින්දු යොදන්න

Replace each set of 3 bits with the relevant octal digit from the above table.
සෑම බිටු 3 කොටසක්ම එයට අදාල අශ්ඨමය ඉලක්කම මගින් ආදේශ කරන්න.

Example 01

0.10110101110_2

0	0	0	.	1	0	1	1	0	1	0	1	1	1	0	0
2^2	2^1	2^0		2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1		4	2	1	4	2	1	4	2	1	4	2	1
0			.	5			5			3			4		

0.5534_8
$0.10110101110_2 = 0.5534_8$

Example 02

101001.1011101_2

1	0	1	0	0	1	.	1	0	1	1	1	0	1	0	0
2^2	2^1	2^0	2^2	2^1	2^0		2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1		4	2	1	4	2	1	4	2	1
5			1			.	5			6			4		

51.564_8
$101001.1011101_2 = 51.564_8$

Example 03

11011.1101011011_2

0	1	1	0	1	1	.	1	1	0	1	0	1	1	0	1	1	0	0
2^2	2^1	2^0	2^2	2^1	2^0		2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1	4	2	1		4	2	1	4	2	1	4	2	1	4	2	1
3			3			.	6			5			5			4		

33.6554_8
$11011.1101011011_2 = 33.6554_8$

S. Converting from Binary to Hexa-decimal
ද්විමය සංඛ්‍යා ෂඩ්දශමය බවට හැරවීම

Split the given binary integer number part from right to left, from the floating point into sets of 4 bits.
දී ඇති ද්විමය පූර්ණ සංඛ්‍යා කොටස දශමස්ථානයේ සිට දකුණේ සිට වමට බිටු 4 කොටස් වලට වෙන්කරන්න.

If necessary, add zeros to the front to complete the 4 bit sets as needed.
බිටු 4 කොටස් සම්පූර්ණ කිරීමට අවශ්‍ය නම් අවශ්‍ය පරිදි මුලට බින්දු යොදන්න

Replace each set of 4 bits with the relevant hexadecimal digit from the above table.
සෑම බිටු 4 කොටසක්ම එයට අදාල ෂඩ්දශමය ඉලක්කම මගින් ආදේශ කරන්න.

Example 01

= $1101110101_2 \rightarrow (\quad)_{16}$

1101110101_2

0	0	1	1	0	1	1	1	0	1	0	1
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1	8	4	2	1
3				7				5			

375_{16}

$1101110101_2 = 375_{16}$

Example 02

= $10110111011_2 \rightarrow (\quad)_{16}$

10110111011_2

0	1	0	1	1	0	1	1	1	0	1	1
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1	8	4	2	1
5				11 (B)				11 (B)			

$5BB_{16}$

$10110111011_2 = 5BB_{16}$

Example 03

= $111101100_2 \rightarrow (\quad)_{16}$

111101100_2

0	0	0	1	1	1	1	0	1	1	0	0
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1	8	4	2	1
1				14 (E)				12 (B)			

$1EB_{16}$

$111101100_2 = 1EB_{16}$

Example 04

= $100101110_2 \rightarrow (\quad)_{16}$

100101110_2

0	0	0	1	0	0	1	0	1	1	1	0
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1	8	4	2	1
1				2				14 (E)			

$12E_{16}$

$100101110_2 = 12E_{16}$

Example 05

= $11010_2 \rightarrow (\quad)_{16}$

11010_2

0	0	0	1	1	0	1	0
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1
1				10 (A)			

$1A_{16}$

$11010_2 = 1A_{16}$

T. Converting Binary floating-point numbers to Hexa-decimal floating-point.
ද්වීමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා ශඩ්දශමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා බවට හැරවීම

Split the given binary floating-point number from left to right, from the floating point into sets of 4 bits.
දී ඇති ද්වීමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා කොටස දශමස්ථානයේ සිට වමේ සිට දකුණට බිටු 4 කොටස් වලට වෙන්කරන්න

Zeros can be added to the end to complete the sets of 4 bits as necessary.
බිටු 4 කොටස් සම්පූර්ණ කිරීමට අවශ්‍ය නම් අවශ්‍ය පරිදි අගට බින්දු යොදන්න

Replace each set of 4 bits with the relevant hexadecimal digit from the above table.
සෑම බිටු 4 කොටසක්ම එයට අදාල ශඩ්දශමය ඉලක්කම මගින් ආදේශ කරන්න

Example 01

0.10110101110_2

0	0	0	0	.	1	0	1	1	0	1	0	1	1	1	0	0
2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1		8	4	2	1	8	4	2	1	8	4	2	1
0				.	11 (B)				5				12 (C)			

$0.B5C_{16}$
$0.10110101110_2 = 0.B5C_{16}$

Example 02

101001.1011101_2

0	0	1	0	1	0	0	1	.	1	0	1	1	1	0	1	0
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1		8	4	2	1	8	4	2	1
2				9				.	11 (B)				10 (A)			

$29.BA_{16}$
$101001.1011101_2 = 29.BA_{16}$

Example 03

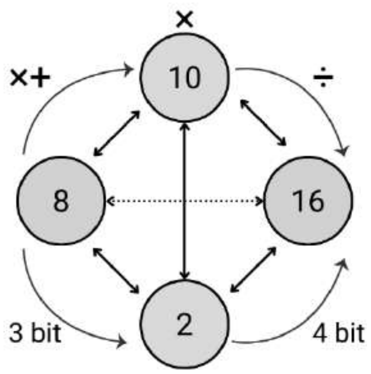
11011.1101011011_2

0	0	0	1	1	0	1	1	.	1	1	0	1	0	1	1	0	1	1	0	0
2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1		8	4	2	1	8	4	2	1	8	4	2	1
1				11 (B)				.	13 (D)				6				12 (C)			

$1B.D6C_{16}$
$11011.1101011011_2 = 1B.D6C_{16}$

Converting Between octal and hexa-decimal
අෂ්ඨමය සහ ෂඩ්දශමය අතර පරිවර්තනය

There is no direct method to convert.
පරිවර්තනය සඳහා සෘජු ක්‍රමයක් නොපවතී



We can use either binary or decimal as an intermediate number system to transform a number among octal and hexa-decimal number systems in 2 steps.
අෂ්ඨමය සහ ෂඩ්දශමය සංඛ්‍යා පද්ධති අතර සංඛ්‍යාවක් පියවර 2කින් පරිවර්තනය කිරීම සඳහා අපට ද්වීමය හෝ දශමය සංඛ්‍යා පද්ධති අතරමැදියෙක් ලෙස භාවිත කළ හැක.

The following methods can be used to convert an octal number into a hexa-decimal number.
අෂ්ඨමය සංඛ්‍යාවක් ෂඩ්දශමය සංඛ්‍යාවක් බවට පරිවර්ථනය කිරීම සඳහා පහත ක්‍රම භාවිතා කල හැක.

1. An octal number can first be converted to a binary number and then the binary number can be converted to a hexa-decimal number.
අෂ්ඨමය සංඛ්‍යාව පලමුව ද්වීමය සංඛ්‍යාවක් බවට පරිවර්ථනය කර ඉන්පසු එම ද්වීමය සංඛ්‍යාව ෂඩ්දශමය සංඛ්‍යාවක් බවට පරිවර්ථනය කල හැක.

2. An octal number can first be converted to a decimal number and then the decimal number can be converted to a hexa-decimal number.
අෂ්ඨමය සංඛ්‍යාව පලමුව දශමය සංඛ්‍යාවක් බවට පරිවර්ථනය කර ඉන්පසු එම දශමය සංඛ්‍යාව ෂඩ්දශමය සංඛ්‍යාවක් බවට පරිවර්ථනය කල හැක.

The following methods can be used to convert a hexa-decimal number into an octal number.
ෂඩ්දශමය සංඛ්‍යාවක් අෂ්ඨමය සංඛ්‍යාවක් බවට පරිවර්ථනය කිරීම සඳහා පහත ක්‍රම භාවිතා කල හැක.

1. A hexa-decimal number can first be converted to a binary number and then the binary number can be converted to an octal number.
ෂඩ්දශමය සංඛ්‍යාව පළමුව ද්වීමය සංඛ්‍යාවක් බවට පරිවර්ථනය කර ඉන්පසු එම ද්වීමය සංඛ්‍යාව අෂ්ඨමය සංඛ්‍යාවක් බවට පරිවර්ථනය කල හැක.

2. The hexa-decimal number can first be converted to a decimal number and then the decimal number can be converted to an octal number.
ෂඩ්දශමය සංඛ්‍යාව පළමුව දශමය සංඛ්‍යාවක් බවට පරිවර්ථනය කර ඉන්පසු එම දශමය සංඛ්‍යාව අෂ්ඨමය සංඛ්‍යාවක් බවට පරිවර්ථනය කල හැක.

U. Converting from octal to Hexa-decimal

අෂ්ටමය සංඛ්‍යා ෂඩ්දශම බවට හැරවීම

1. Convert each octal digit into a 3-digit binary number
එක් එක් අෂ්ටමය ඉලක්කම, ඉලක්කම් 3 හි ද්විමය අංකයක් බවට පරිවර්තනය කරන්න.

2. Combine binary digits into a single sequence. From the right, group them in sets of four.
ද්විමය ඉලක්කම් තනි අනුපිළිවෙලකට ඒකාබද්ධ කරන්න. දකුණේ සිට, ඒවා හතරේ කට්ටලවලට කාණ්ඩ කරන්න.

3. Convert each 4-digit binary group to its hexadecimal equivalent
එක් එක් ඉලක්කම 4 හි ද්විමය කාණ්ඩ එහි ෂඩ්දශමය අගයට පරිවර්තනය කරන්න.

Example 01

$374_8 \rightarrow (\quad)_{16}$

3		
2^2	2^1	2^0
4	2	1
0	1	1

7		
2^2	2^1	2^0
4	2	1
1	1	1

4		
2^2	2^1	2^0
4	2	1
1	0	0

$011\ 111\ 100_2$

11111100_2

1	1	1	1
2^3	2^2	2^1	2^0
8	4	2	1
15 (F)			

1	1	0	0
2^3	2^2	2^1	2^0
8	4	2	1
12 (C)			

FC_{16}

$374_8 = FC_{16}$

Example 02

$253_8 \rightarrow (\quad)_{16}$

2		
2^2	2^1	2^0
4	2	1
0	1	0

5		
2^2	2^1	2^0
4	2	1
1	0	1

3		
2^2	2^1	2^0
4	2	1
0	1	1

$010\ 101\ 011_2$

10101011_2

1	0	1	0
2^3	2^2	2^1	2^0
8	4	2	1
10 (A)			

1	0	1	1
2^3	2^2	2^1	2^0
8	4	2	1
11 (B)			

AB_{16}

$253_8 = AB_{16}$

Example 03

$356_8 \rightarrow (\quad)_{16}$

3		
2^2	2^1	2^0
4	2	1
0	1	1

5		
2^2	2^1	2^0
4	2	1
1	0	1

6		
2^2	2^1	2^0
4	2	1
1	1	0

$011\ 101\ 110_2$

011101110_2

1	1	1	0
2^3	2^2	2^1	2^0
8	4	2	1
14 (E)			

1	1	1	0
2^3	2^2	2^1	2^0
8	4	2	1
14 (E)			

EE_{16}

$356_8 = EE_{16}$

W. Converting from Octal floating-point to Hexa-decimal floating-point

අෂ්ඨමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා ෂඩ්දශමය ඉපිලුම් ලක්ශ්‍ය සංඛ්‍යා බවට හැරවීම

1. For both the integer and fractional parts, convert each octal digit to a 3-digit binary number.
නිඛිල සහ භාගික කොටස් දෙකම සඳහා, එක් එක් අෂ්ඨමය ඉලක්කම, බිටු 3 හි ද්විමය සංඛ්‍යාවකට පරිවර්තනය කරන්න.

2. Combine all binary digits, keeping the decimal point in place. Then, starting from the decimal point, group binary digits into sets of four.
දශම තිත තබා ගනිමින් සියලුම ද්විමය ඉලක්කම් ඒකාබද්ධ කරන්න. ඉන්පසුව, දශම තිතෙන් පටන් ගෙන, ද්විමය ඉලක්කම් හතරක කට්ටලවලට සමූහගත කරන්න.

3. For both integer and fractional parts, convert each 4-digit binary group to its hexadecimal equivalent.
නිඛිල සහ භාගික කොටස් දෙකම සඳහා, එක් එක් ඉලක්කම් 4 හි ද්විමය කාණ්ඩ එහි සමාන ෂඩ්දශමය සංඛ්‍යාවට පරිවර්තනය කරන්න.

Example 01

$0.27_8 \rightarrow (\quad)_{16}$

0			.	2				7		
2^2	2^1	2^0		2^2	2^1	2^0		2^2	2^1	2^0
4	2	1		4	2	1		4	2	1
0	0	0	.	0	1	0		1	1	1
0.010111_2										

0.010111_2													
0	0	0	0	.	0	1	0	1		1	1	0	0
2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0
8	4	2	1		8	4	2	1		8	4	2	1
0				.	5					12 (C)			
$0.5C_{16}$													

Example 02

$1.2_8 \rightarrow (\quad)_{16}$

1			.	2		
2^2	2^1	2^0		2^2	2^1	2^0
4	2	1		4	2	1
0	0	1	.	0	1	0
1.010_2						

1.010_2								
0	0	0	1	.	0	1	0	0
2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0
8	4	2	1		8	4	2	1
1				.	4			
1.4_{16}								

$1.2_8 = 1.4_{16}$

X. Converting from Hexa-decimal to octal

ෂඩ්දශම සංඛ්‍යා අෂ්ඨමය බවට හැරවීම

1. Change each hexadecimal digit into a 4-digit binary number
එක් එක් ෂඩ්දශම ඉලක්කම, ඉලක්කම් 4 හි ද්විමය අංකයක් බවට පරිවර්තනය කරන්න.

2. Combine binary digits into a single sequence. From the right, group them in sets of four.
ද්විමය ඉලක්කම් තනි අනුපිළිවෙලකට ඒකාබද්ධ කරන්න. දකුණේ සිට, ඒවා තුනෙහි කට්ටලවලට කාණ්ඩ කරන්න.

3. Change each 3-digit binary group to its octal equivalent.
එක් එක් ඉලක්කම් 3 හි ද්විමය කාණ්ඩ එහි ෂඩ් අෂ්ඨමය අගයට පරිවර්තනය කරන්න.

Example 01

= $2FA_{16} \rightarrow (\quad)_8$

2			
2^3	2^2	2^1	2^0
8	4	2	1
0	0	1	0

F (15)			
2^3	2^2	2^1	2^0
8	4	2	1
1	1	1	1

A (10)			
2^3	2^2	2^1	2^0
8	4	2	1
1	0	1	0

0010 1111 1010

1011111010_2

0	0	1
2^2	2^1	2^0
4	2	1
1		

0	1	1
2^2	2^1	2^0
4	2	1
3		

1	1	1
2^2	2^1	2^0
4	2	1
7		

0	1	0
2^2	2^1	2^0
4	2	1
2		

1372_8

$2FA_{16} = 1372_8$

Example 02

= $E9_{16} \rightarrow (\quad)_8$

E			
2^3	2^2	2^1	2^0
8	4	2	1
1	1	1	0

9			
2^3	2^2	2^1	2^0
8	4	2	1
1	0	0	1

$1110\ 1001_2$

11101001_2

0	1	1
2^2	2^1	2^0
4	2	1
3		

1	0	1
2^2	2^1	2^0
4	2	1
5		

0	0	1
2^2	2^1	2^0
4	2	1
1		

351_8

$E9_{16} = 351_8$

Y. Converting from Hexa-decimal floating-point to Octal floating-point

ෂඩ්දශමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා අෂ්ඨමය ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා බවට හැරවීම

1. For both the integer and fractional parts, convert each Hexa-decimal digit to a 4-digit binary number.
නිඛිල සහ භාගික කොටස් දෙකම සඳහා, එක් එක් ෂඩ්දශමය ඉලක්ක ම ඉලක්කම් 4 හි ද්විමය අංකයකට පරිවර්තනය කරන්න.

2. Combine all binary digits, keeping the decimal point in place. Then, starting from the decimal point, group binary digits into sets of three.
දශම තිත තබා ගනිමින් සියලුම ද්විමය ඉලක්කම් ඒකාබද්ධ කරන්න. ඉන්පසුව, දශම තිතෙන් පටන් ගෙන, ද්විමය ඉලක්කම් තුනෙහි කට්ටලවලට සමූහගත කරන්න.

3. For both integer and fractional parts, convert each 3-digit binary group to its octal equivalent.
නිඛිල සහ භාගික කොටස් දෙකම සඳහා, එක් එක් ඉලක්කම් 3 හි ද්විමය කාණ්ඩ එහි සමාන අෂ්ඨමය සංඛ්‍යාවට පරිවර්තනය කරන්න.

Example 01

= $0.D1_{16} \rightarrow (\quad)_8$

0				.	D (13)				1			
2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1		8	4	2	1	8	4	2	1
0	0	0	0	.	1	1	0	1	0	0	0	1
0.11010001_2												

0.11010001_2												
0	0	0	.	1	1	0	1	0	0	0	1	0
2^2	2^1	2^0		2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
4	2	1		4	2	1	4	2	1	4	2	1
0			.	6			4			2		
0.642_8												

$0.D1_{16} = 0.642_8$

Example 02

= $F.2_{16} \rightarrow (\quad)_8$

F				.	2			
2^3	2^2	2^1	2^0		2^3	2^2	2^1	2^0
8	4	2	1		8	4	2	1
1	1	1	1	.	0	0	1	0

1111.0010_2

1111.0010_2									
0	0	1	1	1	1	.	0	0	1
2^2	2^1	2^0	2^2	2^1	2^0		2^2	2^1	2^0
4	2	1	4	2	1		4	2	1
1			7			.	1		

17.1_8

$F.2_{16} = 17.1_8$

Most Significant digit (MSD)
වැඩිම වෙසෙසි අංකය

The digit with the highest value in a number is known as Most Significant digit.
සංඛ්‍යාවක වැඩිම අගයක් ඇති ඉලක්කම වැඩිම වෙසෙසි අංකය ලෙස හඳුන්වයි.

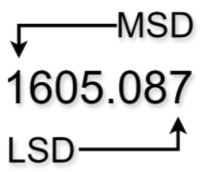
Example:
In 3,472, the 3 on the far left is the MSD, as it has the greatest impact on the number's value.
3,472 හි වම්පස ඇති 3 හි සංඛ්‍යාව MSD වේ, මන්ද එය අංකයේ අගයට විශාලතම බලපෑමක් ඇති කරයි.

Least Significant digit (LSD)
අඩුම වෙසෙසි අංකය

The digit with the lowest value in a number is known as Least Significant digit.
සංඛ්‍යාවක අඩුම අගය ඇති ඉලක්කම අඩුම වෙසෙසි අංකය ලෙස හඳුන්වයි.

Example:
In 3,472 the 2 on the far right is the LSD, as it contributes the smallest impact to the overall value.
3,472 හි, දකුණු කෙළවරේ 2 යනු LSD වේ, එය සමස්ත අගයට කුඩාම භාරයක් දායක වේ.

3,472 හි දකුණු පස ඇති 2 යනු LSD වේ, එය සමස්ත අගයට කුඩාම බලපෑමක් ඇති කරයි.



Understanding MSD and LSD helps in assessing the magnitude of a number and its approximate value.
MSD සහ LSD අවබෝධ කර ගැනීම අංකයක විශාලත්වය සහ එහි ආසන්න අගය තක්සේරු කිරීමට උපකාරී වේ.

Most Significant bit (MSB)
වැඩිම වෙසෙසි බිටුව

The bit with the highest value in a binary number is known as Most Significant bit.
ද්විමය සංඛ්‍යාවක වැඩිම අගයක් ඇති බිටුව වැඩිම වෙසෙසි බිටුව ලෙස හඳුන්වයි.

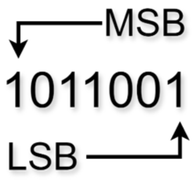
It represents the largest power of 2 in the number.
එය සංඛ්‍යාවේ 2 හි විශාලතම බලය නියෝජනය කරයි.

Example:
In 1011001, the 1 on the far left is the MSB (2^6 position), as it has the greatest impact on the number's value.
1011001 හි වම්පස ඇති 1MSB වේ, මන්ද එය අංකයේ අගයට විශාලතම බලපෑමක් ඇති කරයි.

Least Significant bit (LSB)
අඩුම වෙසෙසි බිටුව

The bit with the lowest value in a binary number is known as LSB.
ද්විමය සංඛ්‍යාවක අඩුම අගය බිටුව LSB ලෙස හඳුන්වයි.

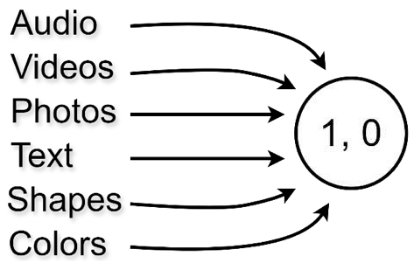
Example:
In 1011001 the 1 on the far right is the LSB, as it contributes the smallest impact to the overall value.
1011001 හි, දකුණු කෙළවරේ 1 යනු LSB වේ,
එය සංඛ්‍යාවේ අගයට අවම බලපෑමක් ඇති කරයි.



Coding Systems
කේත ක්‍රම

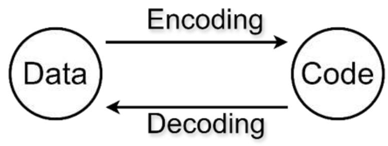
A coding system is a method used to represent data, information, or instructions using specific symbols, numbers, letters, or combinations.
කේතකරණ පද්ධතියක් යනු නිශ්චිත සංකේත, සංඛ්‍යා, අකුරු හෝ සංයෝජන භාවිතා කරමින් දත්ත, තොරතුරු හෝ උපදෙස් නියෝජනය කිරීමට භාවිතා කරන ක්‍රමයකි.

Coding systems make it easier to store, process, and communicate information in computers or other devices.
කේතකරණ පද්ධතියක් පරිගණක හෝ වෙනත් උපාංගවල තොරතුරු ගබඩා කිරීම, සැකසීම සහ සන්නිවේදනය කිරීම පහසු කරයි.



Coding systems allow us to convert data into binary (1s and 0s) so computers can process it. This includes text, numbers, symbols, and multimedia data like images and sounds.
කේතකරණ පද්ධති දත්ත ද්විමය (1 හා 0) බවට පරිවර්තනය කරන අතර පරිගණකවලට ඒවා සැකසීමට ඉඩ සලසයි. මෙයට පාඨ, සංඛ්‍යා, සංකේත, සහ රූප සහ ශබ්ද වැනි බහුමාධ්‍ය දත්ත ඇතුළත් වේ.

Main two concepts in coding system
කේතකරණ පද්ධතියේ ප්‍රධාන සංකල්ප දෙක



- **Encoding**

The process of converting information or data into a code is known as Encoding.
තොරතුරු හෝ දත්ත කේතයක් බවට පරිවර්තනය කිරීමේ ක්‍රියාවලිය කේතනය ලෙස හැඳින්වේ.

Encoding is typically used to ensure data can be properly consumed by different types of systems.
විවිධ වර්ගයේ පද්ධති මගින් දත්ත නිසි ලෙස පරිභෝජනය කළ හැකි බව සහතික කිරීම සඳහා කේතනය කිරීම සාමාන්‍යයෙන් භාවිතා වේ

- **Decoding**

The process of converting the encoded data back to its original format is known as decoding.
කේතනය කරන ලද දත්ත නැවත එහි මුල් ආකෘතියට පරිවර්තනය කිරීමේ ක්‍රියාවලිය විකේතනය ලෙස හැඳින්වේ.

For example, converting text into ASCII or Unicode format ensures that computers understand and display text consistently across different devices and platforms.
උදාහරණයක් ලෙස, පාඨ ASCII හෝ යුනිකෝඩ් ආකෘතියට පරිවර්තනය කිරීම මගින් පරිගණක වලට විවිධ උපාංග සහ වේදිකා හරහා අඛණ්ඩව පාඨ තේරුම් ගැනීම සහ සංදර්ශන කිරීම සහතික කරයි.

1. BCD

Binary
Code
Decimal

BCD (Binary-Coded Decimal) is a coding system where each decimal digit (0–9) is represented by its own binary sequence.
BCD (ද්විමය-කේතිත දශම) යනු එක් එක් දශම ඉලක්කම් (0–9) එහිම ද්විමය අනුක්‍රමයකින් නිරූපණය වන කේතීකරණ පද්ධතියකි.

It takes only 4 bits to store the biggest decimal digit 9 in binary. So, 4 bits should be enough for every other digit.
ද්විමය කේතයක් ලෙස දශමය පාදයේ විශාලතම ඉලක්කම වන 9 ගබඩා කිරීමට අවශ්‍ය වන්නේ බිටු 4ක් පමණි. එබැවින් අනිකුත් සියලු ඉලක්කම් සඳහා බිටු 4ක් ප්‍රමාණවත් විය යුතුය

Therefore, each decimal digit is represented usually by a fixed 4-bit code.
එම නිසා සෑම දශමය ඉලක්කමක්ම නිශ්චිත බිටු 4 ක කේතයක් මගින් නිරූපණය කරයි.

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Empty	1010
Empty	1011
Empty	1100
Empty	1101
Empty	1110
Empty	1111

NOTE
Maximum number of characters that can be represented by using n bits = 2^n
බිටු n ප්‍රමාණයක් භාවිතා කරමින් නිරූපණය කල හැකි උපරිම සංකේත ගණන

Number of maximum characters that can be represented using 4-bit combinations.
බිටු 4 ක සංයෝජන භාවිතා කර නිරූපණය කල හැකි උපරිම සංකේත ප්‍රමාණය
$= 2^4$
$=16$

Number of characters to be represented.
නිරූපණය කිරීමට ඇති සංකේත ප්‍රමාණය
$=10$

Number of remaining places in the code without assigned digits.
එමනිසා සම්බන්ධිත ඉලක්කම් නොමැතිව කේතයේ පවතින හිස්තැන්
$=16 – 10$
$= 6$

NOTE 01:
Only decimal numbers can be converted into BCD. Numbers of other bases should be first converted into base 10 if they are needed to be represented in BCD
BCD බවට පරිවර්තනය කළ හැක්කේ 10 පාදයේ සංඛ්‍යා පමණි. වෙනත් පාද වල සංඛ්‍යා BCD බවට පත් කිරීමට පෙර ඒවා 10 පාදයට පරිවර්තනය කරගත යුතුය.

NOTE 02:
Binary number and the BCD number of the same decimal number are not the same
එකම දහයේ පාදයේ සංඛ්‍යාවේ BCD අගය සහ 2 පාදයේ අගය සමාන නොවේ.

NOTE 03:
Every decimal number is represented by a 4-bit BCD code. But not every 4-bit set is BCD, nor related to a decimal digit.
සෑම දශමය ඉලක්කමක්ම 4-bit BCD කේතයකින් නිරූපණය කෙරේ. නමුත් සෑම 4-bit කට්ටලයක්ම BCD කේතයක් නොවන අතර දශමය ඉලක්කමකට සම්බන්ධ නොවේ.

Converting a given decimal number into a BCD code

දී ඇති දශමය සංඛ්‍යාවක් BCD කේතයක් බවට හැරවීම

Replace each digit of the given decimal number with the relevant 4-bit BCD code.

දී ඇති දශමය සංඛ්‍යාවේ සෑම ඉලක්කමක්ම BCD කේතයට අනුව ඊට අදාල බිටු හතරේ කොටස මගින් ආදේශ කරන්න

Combine all the 4 bit sets together.

එම බිටු හතරේ කොටස් එක ලඟින් ලියන්න

20976_{10}				
2	0	9	7	6
0010	0000	1001	0111	0110
$00100000100101110110_{BCD}$				

Every digit should always be represented by a fixed number of 4 bits, without neglecting any, because it is an encoding-decoding system.

මෙය කේතන-විකේතන පද්ධතියක් වන බැවින් දී ඇති දශමය සංඛ්‍යාවේ සෑම ඉලක්කමක්ම සෑමවිටම කිසිදු බිටුවක් නොසලකා හැරීමකින් තොරව බිටු 4 කොටසකින් නිරූපණය කල යුතුය

NOTE:

Here, zeros in the front or back of the number cannot be neglected

මෙහිදී, සංඛ්‍යාවේ ඉදිරිපස හෝ පිටුපස ඇති ශුන්‍ය අගයන් නොසලකා හැරිය නොහැක.

Example 01:

Convert decimal number 573_{10} to the BCD

573_{10} දශම සංඛ්‍යාව BCD බවට පරිවර්තනය කරන්න

573_{10}		
5	7	3
0101	0111	0011
$0101\ 0111\ 0011_{BCD}$		

$573_{10} = 010101110011_{BCD}$

Example 02:

Convert octal number 275_8 to the BCD

275_8 අෂ්ඨමය සංඛ්‍යාව BCD බවට පරිවර්තනය කරන්න

First, let's convert 275_8 to decimal number

පළමුව, 275_8 දශම සංඛ්‍යාවකට පරිවර්තනය කරමු

275_8		
2×8^2	7×8^1	5×8^0
128	56	5
128 + 56 + 5		
189		
189_{10}		

Now, let's convert 189_{10} to BCD code.

දැන්, 189_{10} BCD කේතයට පරිවර්තනය කරමු.

189_{10}		
1	8	9
0001	0100	0101
000101000101_{BCD}		

NOTE 01:

Related Concepts to BCD

Packed Decimal Representation is a form of BCD where two decimal digits are packed into a single byte.

Packed Decimal Representation යනු දශම ඉලක්කම් දෙකක් තනි බයිටයකට අසුරා ඇති BCD ආකාරයකි.

NOTE 02:

BCD as a concept doesn't have a single inventor. It was a natural evolution.

සංකල්පයක් ලෙස BCD හට තනි නිපැයුම්කරුවෙකු නොමැත. එය ස්වභාවික පරිණාමයකි.

Advantages of BCD
BCD වල වාසි

Since each decimal digit is represented independently here, BCD representations are easier for humans to read and understand compared to binary numbers.
මෙහි එක් එක් දශමය ඉලක්කම් ස්වාධීනව නිරූපණය වන බැවින්, BCD නිරූපණයන් මිනිසුන්ට කියවීමට සහ තේරුම් ගැනීමට පහසු වේ.

Arithmetic operations on BCD numbers can be performed using techniques similar to decimal arithmetic.
දශමය ගණිතයට සමාන ශිල්පීය ක්‍රම භාවිතයෙන් BCD සංඛ්‍යා මත අංක ගණිත මෙහෙයුම් සිදු කළ හැක.

The format of a number can be easily interchanged from Decimal to BCD rather than from Decimal to binary.
සංඛ්‍යාවක ආකෘතිය දශමයේ සිට ද්විමය දක්වා හුවමාරු කිරීමට වඩා පහසුවෙන් දශමයේ සිට BCD දක්වා හුවමාරු කළ හැක.

BCD can represent decimal values without rounding errors, which occur when converting between binary and decimal in floating-point arithmetic. Therefore, it is used in financial computing and databases where decimal accuracy is crucial.
ඉපිලුම් ලක්ෂ්‍ය අංක ගණිතයේ ද්විමය සහ දශමය අතර පරිවර්තනය කිරීමේදී මෙන් වැටයීමේ දෝෂ වලින් තොරව BCD මගින් ඉපිලුම් ලක්ෂ්‍ය අගයන් නිරූපණය කළ හැක. එබැවින් එය දශමය නිරවද්‍යතාව තීරණාත්මක වන දත්ත සමුදාය වලදී සහ මූල්‍ය පරිගණනයේ දී භාවිතා වේ.

Addition and subtraction can be easier with BCD than with binary representations.
එකතු කිරීම් සහ අඩු කිරීම් ද්විමය පාදයෙන් නිරූපණය කිරීමට වඩා BCD සමඟ පහසුය.

BCD can be implemented efficiently in hardware using specialized circuits.
විශේෂිත පරිපථ භාවිතයෙන් දෘඩාංගවල BCD කාර්යක්ෂමව ක්‍රියාත්මක කළ හැක.

Disadvantages of BCD
BCD වල අවාසි

BCD is less efficient in terms of storage compared to pure binary, as each decimal digit is stored separately using 4 bits, even though only 10 values (0–9) are possible.
සෑම දශමය ඉලක්කමක්ම බිටු 4ක් භාවිතා කරමින් වෙන වෙනම ගබඩා කර ඇති බැවින්, අගයන් 10ක් (0–9) පමණක් කළ හැකි වුවද, පූර්ණ ද්විමය හා සසඳන විට ගබඩා කිරීමේදී BCD අඩු කාර්යක්ෂම වේ.

BCD requires more bits to represent a decimal number compared to binary.
ද්විමය හා සසඳන විට දශමය සංඛ්‍යාවක් නිරූපණය කිරීමට BCD හට වැඩි බිටු අවශ්‍ය වේ.

For example, representing the decimal number 15 requires 8 bits in BCD, while it only requires 4 bits in binary.
උදාහරණයක් ලෙස, දශම අංක 15 නියෝජනය කිරීම සඳහා BCD හි බිටු 8ක් අවශ්‍ය වන අතර, එයට අවශ්‍ය වන්නේ ද්විමය වශයෙන් බිටු 4ක් පමණි.

15_{10} = 4 bits (1111_2)
15_{10} = 8 bits (00001101_{BCD})

In modern general-purpose computing, binary representations are more standard, making BCD less common and outdated.
නූතන පොදු කාර්ය පරිගණනයේ දී, ද්විමය නිරූපණයන් වඩාත් සම්මත වන අතර, BCD යනු භාවිතය අඩු සහ යල් පැන ගිය එකකි.

Multiplying and dividing BCD code is more complex than with pure binary numbers.
BCD කේත ගුණ කිරීම සහ බෙදීම සම්මත ද්විමය සංඛ්‍යා වලදී ට වඩා සංකීර්ණ වේ.

Applications of BCD
BCD හි යෙදුම්

BCD is commonly used in calculators to represent numbers and perform arithmetic operations.
BCD සාමාන්‍යයෙන් සංඛ්‍යා නිරූපණය කිරීමට සහ අංක ගණිත මෙහෙයුම් සිදු කිරීමට ගණක යන්ත්‍රවල භාවිතා වේ.

BCD is used to display time and date in digital devices.
අංකිත උපාංගවල වේලාව සහ දිනය පෙන්වීමට BCD භාවිතා කරයි.

Some computer peripherals, such as keyboards and displays, may use BCD to represent data.
යතුරුපුවරු සහ සංදර්ශක වැනි සමහර පරිගණක උපාංග, දත්ත නිරූපණය කිරීමට BCD භාවිතා කළ හැක.

Floating-point numbers are not directly compatible with BCD because BCD focuses only on integer digits. If you need to work with floating-point numbers, using systems like IEEE 754 is more appropriate.
BCD පූර්ණ සංඛ්‍යා ඉලක්කම් මත පමණක් අවධානය යොමු කරන නිසා පාවෙන ලක්ෂ්‍ය සංඛ්‍යා BCD සමඟ සෘජුව නොගැළපේ. ඔබට පාවෙන ලක්ෂ්‍ය අංක සමඟ වැඩ කිරීමට අවශ්‍ය නම්, IEEE 754 වැනි පද්ධති භාවිතා කිරීම වඩාත් සුදුසුය.

2. ASCII

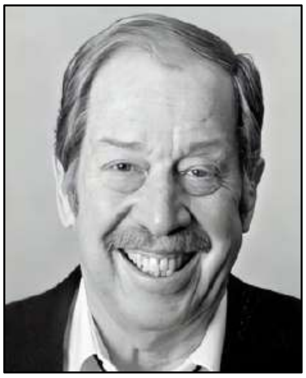
American
Standard
Code for
Information
Interchange

ASCII emerged in the 1960s to standardize text representation using binary code
ASCII, 1960 දී ද්විමය කේතය භාවිතයෙන් පාඨ නිරූපණය ප්‍රමිතිකරණය කිරීමට මතු විය.

Like Morse code, Telegraphic codes were among the earliest examples of character encoding, but they were designed for specific communication systems.
මෝර්ස් කේතය මෙන්, ටෙලිග්‍රාෆික් කේත අක්ෂර කේතකරණ පද්ධතිය පැරණිතම උදාහරණ අතර විය, නමුත් ඒවා විශේෂිත සන්නිවේදන පද්ධති සඳහා නිර්මාණය කර ඇත.

In 1963, the ASA (later renamed the American National Standards Institute, ANSI) established a committee to standardize the 7-bit code.
1963 දී, ASA (පසුව American National Standards Institute, ANSI ලෙස නම් කරන ලදී) 7-bit කේතය ප්‍රමිතිකරණය කිරීමට කමිටුවක් පිහිටුවන ලදී.

ASCII was officially developed in 1963 under the direction of Bob Bemer, who led the creation of the code to standardize text encoding for use in computers and communication equipment.
ASCII නිල වශයෙන් 1963 Bob Bemer ගේ මඟපෙන්වීම යටතේ සංවර්ධනය කරන ලද අතර, පරිගණක සහ සන්නිවේදන උපකරණවල භාවිතය සඳහා පාඨ කේතනය ප්‍රමිතිගත කිරීම සඳහා කේතය නිර්මාණය කිරීමට නායකත්වය දුන්නේය.



Later became the default standard for text files in UNIX and DOS systems.
පසුව UNIX සහ DOS පද්ධතිවල අක්ෂර ගොනු හැසිරවීම සඳහා ප්‍රමිතිය බවට පත් විය.

The first version of ASCII included 7-bit codes, allowing for 128 different characters, including control characters (for commands like carriage return, newline, and tab) and printable characters (letters, numbers, symbols).

ASCII හි පළමු අනුවාදයට 7-බිටු කේත ඇතුළත් වූ අතර, පාලන අනුලක්ෂණ (කැරේජ් රිටර්න්, නිව්ලයින් සහ ටැබ් වැනි විධාන සඳහා) සහ මුද්‍රණය කළ හැකි අනුලක්ෂණ (අකුරු, අංක, සංකේත) ඇතුළුව විවිධ අක්ෂර 128 සඳහා ඉඩ ලබා දේ.

Numeric සංඛ්‍යාංක 0 - 9
Alphabetic අක්ෂරමය a - z A - Z
Special විශේෂ # @ ! . + -

ASCII does not represent world languages.
ASCII ලෝක භාෂාවන් නියෝජනය නොකරයි

Extended ASCII uses 8 bits, allowing for 256 characters (0–255). The extra 128 characters (128–255) include symbols, foreign language characters, and graphical characters. But it is not standardized.

Extended ASCII අනුලක්ෂණ 256 (0–255) සඳහා ඉඩ දෙමින් බිටු 8ක් භාවිතා කරයි. අමතර අක්ෂර 128 (128–255) තුළ සංකේත, විදේශීය භාෂා අක්ෂර සහ චිත්‍රක අක්ෂර ඇතුළත් වේ. නමුත් එය සම්මත ක්‍රමයක් නොවේ.

Number of maximum characters that can be represented by 7 bit combinations
බිටු 7 ක සංයෝජන භාවිතා කර නිරූපණය කලහැකි උපරිම සංකේත ප්‍රමාණය

$= 2^7 = 128$

Number of maximum characters that can be represented by 8 bit combinations
බිටු 8 ක සංයෝජන භාවිතා කර නිරූපණය කලහැකි උපරිම සංකේත ප්‍රමාණය
$= 2^8 = 25$

ASCII uses a linear representation of letters in the order of the alphabet.
ASCII කේත ක්‍රමය හෝඩියේ අනුපිළිවෙලට අක්ෂර වල රේඛීය නිරූපණයක් භාවිතා කරයි.

It matches newer coding systems and has even become a subset of them.
එය නවීන කේතකරණ පද්ධති සමඟ ගැලපෙන අතර ඒවායේ උප කුලකයක් බවට පත් වී ඇත.

character	Decimal	ASCII
	0	0000000
0	48	0110000
9	57	0111001
A	65	1000001
B	66	1000010
a	97	1100001
b	98	1100010
	128	1111111

ASCII Control Characters

DECI	BINARY	CHAR	DESCRIPTION
0	00000000	NUL	Null char
1	00000001	SOH	Start of Heading
2	00000010	STX	Start of Text
3	00000011	ETX	End of Text
4	00000100	EOT	End of Transmission
5	00000101	ENQ	Enquiry
6	00000110	ACK	Acknowledgment
7	00000111	BEL	Bell
8	00001000	BS	Back Space
9	00001001	HT	Horizontal Tab
10	00001010	LF	Line Feed
11	00001011	VT	Vertical Tab
12	00001100	FF	Form Feed
13	00001101	CR	Carriage Return
14	00001110	SO	Shift Out / X-On
15	00001111	SI	Shift In / X-Off
16	00010000	DLE	Data Line Escape
17	00010001	DC1	Device Control 1 (oft. XON)
18	00010010	DC2	Device Control 2
19	00010011	DC3	Device Control 3 (oft. XOFF)
20	00010100	DC4	Device Control 4
21	00010101	NAK	Negative Acknowledgement

22	00010110	SYN	Synchronous Idle
23	00010111	ETB	End of Transmit Block
24	00011000	CAN	Cancel
25	00011001	EM	End of Medium
26	00011010	SUB	Substitute
27	00011011	ESC	Escape
28	00011100	FS	File Separator
29	00011101	GS	Group Separator
30	00011110	RS	Record Separator
31	00011111	US	Unit Separator

ASCII printable characters

DECIMAL	BINARY	CHAR	DESCRIPTION
32	00100000		Space
33	00100001	!	Exclamation
34	00100010	"	Double quotes
35	00100011	#	Number
36	00100100	$	Dollar
37	00100101	%	Per cent sign
38	00100110	&	Ampersand
39	00100111	'	Single quote
40	00101000	(	open bracket
41	00101001	)	close bracket
42	00101010	*	Asterisk
43	00101011	+	Plus
44	00101100	,	Comma
45	00101101	-	Hyphen
46	00101110	.	Dot or full stop
47	00101111	/	Slash or divide
48	00110000	0	Zero
49	00110001	1	One
50	00110010	2	Two
51	00110011	3	Three
52	00110100	4	Four
53	00110101	5	Five
54	00110110	6	Six
55	00110111	7	Seven
56	00111000	8	Eight
57	00111001	9	Nine
58	00111010	:	Colon
59	00111011	;	Semicolon
60	00111100	<	Less than
61	00111101	=	Equals
62	00111110	>	Greater than
63	00111111	?	Question mark
64	01000000	@	At symbol
65	01000001	A	Uppercase A
66	01000010	B	Uppercase B
67	01000011	C	Uppercase C
68	01000100	D	Uppercase D
69	01000101	E	Uppercase E
70	01000110	F	Uppercase F
71	01000111	G	Uppercase G
72	01001000	H	Uppercase H
73	01001001	I	Uppercase I
74	01001010	J	Uppercase J
75	01001011	K	Uppercase K
76	01001100	L	Uppercase L
77	01001101	M	Uppercase M
78	01001110	N	Uppercase N
79	01001111	O	Uppercase O
80	01010000	P	Uppercase P
81	01010001	Q	Uppercase Q
82	01010010	R	Uppercase R
83	01010011	S	Uppercase S
84	01010100	T	Uppercase T
85	01010101	U	Uppercase U
86	01010110	V	Uppercase V
87	01010111	W	Uppercase W
88	01011000	X	Uppercase X
89	01011001	Y	Uppercase Y
90	01011010	Z	Uppercase Z
91	01011011	[	Open bracket
92	01011100	\	Backslash
93	01011101	]	Close bracket
94	01011110	^	Caret
95	01011111	_	Underscore
96	01100000	`	Grave accent
97	01100001	a	Lowercase a
98	01100010	b	Lowercase b
99	01100011	c	Lowercase c
100	01100100	d	Lowercase d
101	01100101	e	Lowercase e
102	01100110	f	Lowercase f
103	01100111	g	Lowercase g
104	01101000	h	Lowercase h
105	01101001	i	Lowercase i
106	01101010	j	Lowercase j
107	01101011	k	Lowercase k
108	01101100	l	Lowercase l
109	01101101	m	Lowercase m
110	01101110	n	Lowercase n
111	01101111	o	Lowercase o
112	01110000	p	Lowercase p
113	01110001	q	Lowercase q
114	01110010	r	Lowercase r
115	01110011	s	Lowercase s
116	01110100	t	Lowercase t
117	01110101	u	Lowercase u
118	01110110	v	Lowercase v
119	01110111	w	Lowercase w

120	01111000	x	Lowercase x
121	01111001	y	Lowercase y
122	01111010	z	Lowercase z
123	01111011	{	Open brace
124	01111100	|	Vertical bar
125	01111101	}	Close brace
126	01111110	~	tilde
127	01111111		Delete

Extended 8-bit ASCII

128	10000000	€	Euro sign
129	10000001		
130	10000010	‚	Single low-9 quotation
131	10000011	ƒ	f with hook
132	10000100	„	Double low-9 quotation
133	10000101	…	Horizontal ellipsis
134	10000110	†	Dagger
135	10000111	‡	Double dagger
136	10001000	ˆ	Modifier circumflex
137	10001001	‰	Per mille sign
138	10001010	Š	S with caron
139	10001011	‹	Single left-pointing angle quotation
140	10001100	Œ	ligature OE
141	10001101		
142	10001110	Ž	Z with caron
143	10001111		
144	10010000		
145	10010001	‘	Left single quotation
146	10010010	’	Right single quotation
147	10010011	“	Left double quotation
148	10010100	”	Right double quotation
149	10010101	•	Bullet
150	10010110	–	En dash
151	10010111	—	Em dash
152	10011000	˜	tilde
153	10011001	™	Trade mark sign
154	10011010	š	S with caron
155	10011011	›	right-pointing angle quotation mark
156	10011100	œ	ligature oe
157	10011101		
158	10011110	ž	z with caron
159	10011111	Ÿ	Y with diaeresis
160	10100000		Non-breaking space
161	10100001	¡	Inverted exclamation
162	10100010	¢	Cent sign
163	10100011	£	Pound sign
164	10100100	¤	Currency sign
165	10100101	¥	Yen sign
166	10100110	¦	Pipe, Broken vertical bar
167	10100111	§	Section sign
168	10101000	¨	Spacing diaeresis - umlaut
169	10101001	©	Copyright sign
170	10101010	ª	Feminine ordinal indicator
171	10101011	«	Left double angle quotes
172	10101100	¬	Not sign
173	10101101		Soft hyphen
174	10101110	®	Registered trademark
175	10101111	¯	Spacing macron - overline
176	10110000	°	Degree sign
177	10110001	±	Plus-or-minus sign
178	10110010	²	Superscript two
179	10110011	³	Superscript three
180	10110100	´	Acute accent - spacing acute
181	10110101	µ	Micro sign
182	10110110	¶	Pilcrow sign - paragraph sign
183	10110111	·	Middle dot - Georgian comma
184	10111000	¸	Spacing cedilla
185	10111001	¹	Superscript one
186	10111010	º	Masculine ordinal indicator
187	10111011	»	Right double angle quotes
188	10111100	¼	Fraction one quarter
189	10111101	½	Fraction one half
190	10111110	¾	Fraction three quarters
191	10111111	¿	Inverted question mark
192	11000000	À	A with grave
193	11000001	Á	A with acute
194	11000010	Â	A with circumflex
195	11000011	Ã	A with tilde
196	11000100	Ä	A with diaeresis
197	11000101	Å	A with ring above
198	11000110	Æ	AE
199	11000111	Ç	C with cedilla
200	11001000	È	E with grave
201	11001001	É	E with acute
202	11001010	Ê	E with circumflex
203	11001011	Ë	E with diaeresis
204	11001100	Ì	I with grave
205	11001101	Í	I with acute
206	11001110	Î	I with circumflex
207	11001111	Ï	I with diaeresis
208	11010000	Ð	ETH
209	11010001	Ñ	N with tilde
210	11010010	Ò	O with grave
211	11010011	Ó	O with acute
212	11010100	Ô	O with circumflex
213	11010101	Õ	O with tilde
214	11010110	Ö	O with diaeresis

215	11010111	×	Multiplication sign
216	11011000	Ø	O with slash
217	11011001	Ù	U with grave
218	11011010	Ú	U with acute
219	11011011	Û	U with circumflex
220	11011100	Ü	U with diaeresis
221	11011101	Ý	Y with acute
222	11011110	Þ	THORN
223	11011111	ß	sharp s - ess-zed
224	11100000	à	a with grave
225	11100001	á	a with acute
226	11100010	â	a with circumflex
227	11100011	ã	a with tilde
228	11100100	ä	a with diaeresis
229	11100101	å	a with ring above
230	11100110	æ	ae
231	11100111	ç	c with cedilla
232	11101000	è	e with grave
233	11101001	é	e with acute
234	11101010	ê	e with circumflex
235	11101011	ë	e with diaeresis
236	11101100	ì	i with grave
237	11101101	í	i with acute
238	11101110	î	i with circumflex
239	11101111	ï	i with diaeresis
240	11110000	ð	eth
241	11110001	ñ	n with tilde
242	11110010	ò	o with grave
243	11110011	ó	o with acute
244	11110100	ô	o with circumflex
245	11110101	õ	o with tilde
246	11110110	ö	o with diaeresis
247	11110111	÷	Division sign
248	11111000	ø	o with slash
249	11111001	ù	u with grave
250	11111010	ú	u with acute
251	11111011	û	u with circumflex
252	11111100	ü	u with diaeresis
253	11111101	ý	y with acute
254	11111110	þ	thorn
255	11111111	ÿ	y with diaeresis

FUN FACT: ASCII Arts

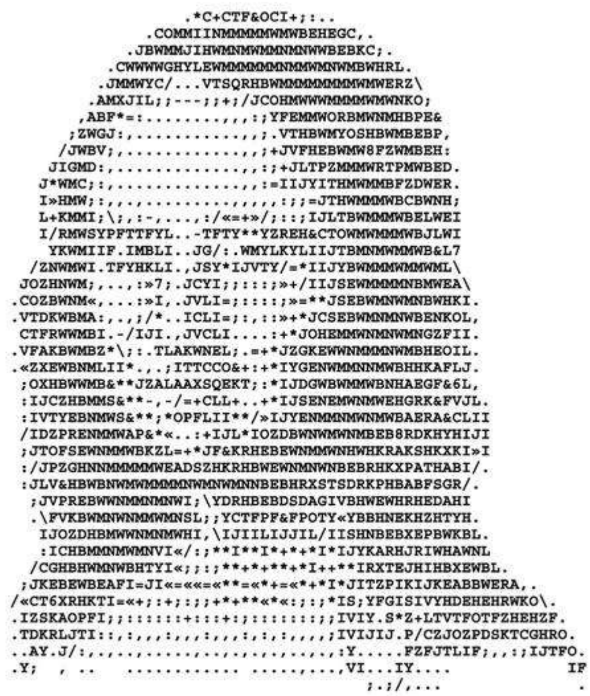
¯_(ツ)_/¯

ASCII arts is a graphic design technique that uses printable characters from the ASCII standard to generate artistic images.

ASCII කලා යනු කලාත්මක රූප ජනනය කිරීම සඳහා ASCII ප්‍රමිතියෙන් මුද්‍රණය කළ හැකි අක්ෂර භාවිතා කරන ග්‍රැෆික් නිර්මාණ තාක්‍ෂණයකි.

The "@" symbol had little use before email. ASCII led it to become the designator in email addresses.

ඊමේල් භාවිතයට පෙර "@" සංකේතයට එතරම් අරමුණක් තිබුණේ නැත. ASCII කේත ක්‍රමය එය ඊමේල් ලිපිනවල භාවිතයට මඟ පෑදීය.



Arabic

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8x																
9x																
Ax	NB SP				¤								،	SH Y		
Bx												؛				؟
Cx		ء	آ	أ	ؤ	إ	ئ	ا	ب	ة	ت	ث	ج	ح	خ	د
Dx	ذ	ر	ز	س	ش	ص	ض	ط	ظ	ع	غ					
Ex	ـ	ف	ق	ك	ل	م	ن	ه	و	ى	ي	◌ً	◌ٌ	◌ٍ	◌َ	◌ُ
Fx	◌ِ	◌ّ	◌ْ													

Greek

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8x																
9x																
Ax	NB SP	‘	’	£	€	₯	¦	§	¨	©	ͺ	«	¬	SH Y		―
Bx	°	±	²	³	΄	΅	Ά	·	Έ	Ή	Ί	»	Ό	½	Ύ	Ώ
Cx	ΐ	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
Dx	Π	Ρ		Σ	Τ	Υ	Φ	Χ	Ψ	Ω	Ϊ	Ϋ	ά	έ	ή	ί
Ex	ΰ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
Fx	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	ϊ	ϋ	ό	ύ	ώ	

Western Europe

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8x																
9x																
Ax	NB SP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	SH Y	®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Thailand

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8x																
9x																
Ax	NB SP	ก	ข	ฃ	ค	ฅ	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ
Bx	ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ
Cx	ภ	ม	ย	ร	ฤ	ล	ฦ	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	ฯ
Dx	ะ	◌ั	า	◌ำ	◌ิ	◌ี	◌ึ	◌ื	◌ุ	◌ู	◌ฺ					฿
Ex	เ	แ	โ	ใ	ไ	ๅ	ๆ	◌็	◌่	◌้	◌๊	◌๋	◌์	◌ํ	◌๎	๏
Fx	๐	๑	๒	๓	๔	๕	๖	๗	๘	๙	๚	๛				

Advantages of ASCII
ASCII වල වාසි

Easy to understand and implement.
තේරුම් ගැනීමට සහ ක්‍රියාත්මක කිරීමට පහසුය.

Compatible for almost all systems and devices can process ASCII.
සියලුම පද්ධති සහ උපාංග සඳහා ASCII සැකසීමට ඇති හැකියාව.

Disadvantages of ASCII
ASCII වල අවාසි

ASCII can only represent 128 (or 256 in extended ASCII) characters, which is not sufficient for languages with large alphabets or symbols (e.g., Chinese, Arabic).
ASCII හට නිරූපණය කළ හැක්කේ විශාල හෝඩියක් හෝ සංකේත (උදා: චීන, අරාබි) සහිත භාෂා සඳහා ප්‍රමාණවත් නොවන අනුලක්ෂණ 128ක් (හෝ extended ASCII හි 256ක්) පමණි.

Uses of ASCII
ASCII භාවිතය

ASCII is used to encode text in simple text files (.txt), allowing computers to display readable characters.
ASCII සරල පාඨ ගොනු (.txt) තුළ පාඨ කේතනය කිරීමට භාවිතා කරයි, කියවිය හැකි අනුලක්ෂණ පෙන්වීමට පරිගණකවලට ඉඩ සලසයි.

Used in many programming languages to represent characters and control characters.
බොහෝ ක්‍රමලේඛන භාෂාවල අනුලක්ෂණ නියෝජනය කිරීමට සහ අනුලක්ෂණ පාලනය කිරීමට භාවිතා කරයි.

ASCII is used in email protocols, URLs, and other forms of digital communication where text needs to be encoded.
ASCII විද්‍යුත් තැපැල් නියමාවලිය, URL සහ පාඨ සංකේතනය කළ යුතු වෙනත් ආකාරයේ අංකිත සන්නිවේදනයන්හි භාවිතා වේ.

3. EBCDIC

Extended
Binary
Coded
Decimal
Interchange
Code

EBCDIC is a character encoding system developed by IBM in the 1960s.
EBCDIC යනු 1960 දී IBM විසින් සංවර්ධනය කරන ලද අක්ෂර කේතකරණ පද්ධතියකි.

It was primarily used in IBM mainframes and other IBM systems, serving as an alternative to ASCII.
එය මූලික වශයෙන් IBM mainframe සහ අනෙකුත් IBM පද්ධතිවල භාවිතා කරන ලද අතර, ASCII සඳහා විකල්පයක් ලෙස සේවය කරයි.

Number of maximum characters that can be represented by 8-bit combinations.
බිටු 8 ක සංයෝජන භාවිතා කර නිරූපණය කලහැකි උපරිම සංකේත ප්‍රමාණය

$= 2^8 = 256$

EBCDIC is not compatible with modern coding systems like ASCII and Unicode
ASCII සහ Unicode ආදී නූතන කේත ක්‍රම වලට සහයෝගය නොදක්වයි

Does not represent a linear order of characters. So, characters cannot be predicted according to next or past.
අක්ෂරවල රේඛීය නිරූපණයක් නොපවතී. එමනිසා පෙර සහ පසු සංකේත මගින් වෙනත් සංකේත පුරෝකථනය කිරීම සිදුකල නොහැක

Key Points about EBCDIC
EBCDIC පිළිබඳ මූලික කරුණු

EBCDIC supports a wide range of characters, including letters, numbers, punctuation marks, and special symbols.
EBCDIC අකුරු, අංක, විරාම ලකුණු සහ විශේෂ සංකේත ඇතුළු පුළුල් පරාසයක අක්ෂර සඳහා සහය දක්වයි.

While it's not directly compatible with ASCII, there are conversion methods to translate between the two.
එය ASCII සමඟ සෘජුව නොගැළපෙන නමුත්, දෙක අතර පරිවර්තනය කිරීමට පරිවර්තන ක්‍රම තිබේ.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	SEL	HT	RNL	DEL	GE	SPS	RPT	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	RES/ENP	NL	BS	POC	CAN	EM	UBS	CU1	IFS	IGS	IRS	IUS/ITB
2	DS	SOS	FS	WUS	BYP/INP	LF	ETB	ESC	SA	SFE	SM/SW	CSP	MFA	ENQ	ACK	BEL
3			SYN	IR	PP	TRN	NBS	EOT	SBS	IT	RFF	CU3	DC4	NAK		SUB
4	SP										¢	.	<	(	+	|
5	&										!	$	*	)	;	¬
6	-	/									¦	,	%	_	>	?
7										`	:	#	@	'	=	"
8		a	b	c	d	e	f	g	h	i						±
9		j	k	l	m	n	o	p	q	r						
A		~	s	t	u	v	w	x	y	z						
B	^										[	]				
C	{	A	B	C	D	E	F	G	H	I						
D	}	J	K	L	M	N	O	P	Q	R						
E	\		S	T	U	V	W	X	Y	Z						
F	0	1	2	3	4	5	6	7	8	9						EO

EBCDIC vs ASCII Encoding Systems Difference
EBCDIC එදිරිව ASCII කේතන පද්ධතියේ වෙනස

Character	EBCDIC Binary	EBCDIC Hexa	ASCII Binary	ASCII Hexa
A	1100 0001	C1	0100 0001	41
B	1100 0010	C2	0100 0010	42
a	1000 0001	81	0110 0001	61
b	1000 0010	82	0110 0010	62
1	1111 0001	F1	0011 0001	31
2	1111 0010	F2	0011 0010	32

CAPITAL LETTERS

Character	Hex Code	Binary
A	C1	1100 0001
B	C2	1100 0010
C	C3	1100 0011
D	C4	1100 0100
E	C5	1100 0101
F	C6	1100 0110
G	C7	1100 0111
H	C8	1100 1000
I	C9	1100 1001
J	D1	1101 0001
K	D2	1101 0010
L	D3	1101 0011
M	D4	1101 0100
N	D5	1101 0101
O	D6	1101 0110
P	D7	1101 0111
Q	D8	1101 1000
R	D9	1101 1001
S	E2	1110 0010
T	E3	1110 0011
U	E4	1110 0100
V	E5	1110 0101
W	E6	1110 0110
X	E7	1110 0111
Y	E8	1110 1000
Z	E9	1110 1001

SIMPLE LETTERS

Character	Hex Code	Binary
a	81	1000 0001
b	82	1000 0010
c	83	1000 0011
d	84	1000 0100
e	85	1000 0101
f	86	1000 0110
g	87	1000 0111
h	88	1000 1000
i	89	1000 1001
j	91	1001 0001
k	92	1001 0010
l	93	1001 0011
m	94	1001 0100
n	95	1001 0101
o	96	1001 0110
p	97	1001 0111
q	98	1001 1000
r	99	1001 1001
s	A2	1010 0010
t	A3	1010 0011
u	A4	1010 0100
v	A5	1010 0101
w	A6	1010 0110
x	A7	1010 0111
y	A8	1010 1000
z	A9	1010 1001

Character	Hex Code	Binary
0	F0	1111 0000
1	F1	1111 0001
2	F2	1111 0010
3	F3	1111 0011
4	F4	1111 0100
5	F5	1111 0101
6	F6	1111 0110
7	F7	1111 0111
8	F8	1111 1000
9	F9	1111 1001

+	4E	0100 1110
-	60	0110 0000
.	4B	0100 1011
/	61	0110 0001
,	6B	0110 1011
(	4D	0100 1101
)	5D	0101 1101
&	50	0101 0000
!	5A	0101 1010
$	5B	0101 1011
#	7B	0111 1011
@	7C	0111 1100
%	6C	0110 1100
"	7D	0111 1101
:	7A	0111 1010
?	6F	0110 1111

Advantages of EBCDIC
EBCDIC හි වාසි

With fewer bits needed to represent a character, the code is more efficient.
අනුලක්ෂණයක් නිරූපණය කිරීමට අවශ්‍ය බිටු ගණන අඩුවන අතර, කේතය වඩාත් කාර්යක්ෂම වේ.

Unlike ASCII, it includes built-in error-checking capabilities.
ASCII මෙන් නොව, එහි දෝෂ පිරික්සීමේ හැකියාවන්ද ඇතුළත් වේ.

Disadvantages of EBCDIC
EBCDIC හි අවාසි

There is fewer software available that can work with it because it is not as commonly used as ASCII.
එය ASCII තරම් බහුලව භාවිතා නොවන නිසා එය සමඟ වැඩ කළ හැකි මෘදුකාංග අඩුයි.

Data alteration between EBCDIC and ASCII is more challenging than it is with ASCII.
EBCDIC සහ ASCII අතර දත්ත වෙනස් කිරීම ASCII සමඟ වඩා අභියෝගාත්මක ය.

Other computer makers are incapable to use EBCDIC without a license from IBM since it is a proprietary code that the company created.
වෙනත් පරිගණක නිෂ්පාදකයින්ට IBM හි බලපත්‍රයක් නොමැතිව EBCDIC භාවිතා කිරීමට නොහැකි වන්නේ එය සමාගම විසින් නිර්මාණය කරන ලද හිමිකාර කේතයක් වන බැවිනි.

Though EBCDIC is not common in new systems, its continued presence in mainframes, financial systems, and legacy applications makes it essential in certain enterprise and government infrastructures.
නව පද්ධතිවල EBCDIC සුලභ නොවුවත්, එය ප්‍රධාන රාමු, මූල්‍ය පද්ධති සහ උරුම යෙදුම්වල අඛණ්ඩව පැවතීම ඇතැම් ව්‍යවසාය සහ රජයේ යටිතල ව්‍යුහයන් තුළ එය අත්‍යවශ්‍ය වේ.

4. Unicode

Unicode is a universal character encoding standard that provides a unique number for every character across languages and scripts, making almost all characters accessible across platforms, programs, and devices.
යුනිකෝඩ් යනු භාෂා සහ ස්ක්‍රිප්ට් හරහා සෑම අක්ෂරයකටම අනන්‍ය අංකයක් සපයන විශ්වීය අක්ෂර කේතකරණ ප්‍රමිතියකි, වේදිකා, වැඩසටහන් සහ උපාංග හරහා සියලුම අක්ෂර පසාම ප්‍රවේශ විය හැක.

Unicode was first created by Joe Becker from Xerox in 1987. The first version was released in October 1991 with 7,161 characters.
යුනිකෝඩ් මුලින්ම නිර්මාණය කළේ Xerox ආයතනයේ ජෝ බෙකර් විසින් 1987 දීය. පළමු අනුවාදය 1991 ඔක්තෝම්බර් මාසයේදී අක්ෂර 7161 කින් නිකුත් කරන ලදී.

xerox

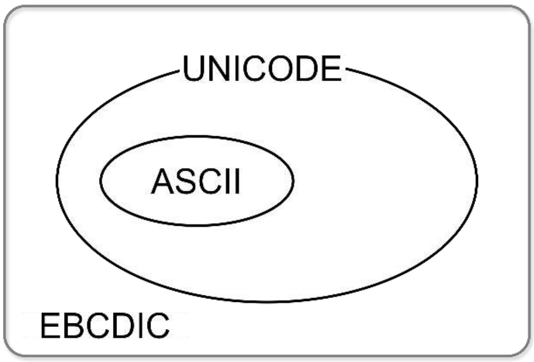
The Unicode Consortium, the non-profit organization that coordinates Unicode's development which includes full time member companies like Adobe, Airbnb, Amazon, Apple, Meta, Google, Huawei, IBM, Microsoft, Netflix, Oracle, and Ministry of Electronics and Information Technology India, and the University of California.
යුනිකේත සම්මේලනය යනු ඉහත නම් සඳහන් ආයතන සහ තවත් ආයතන පවතින Unicode සංවර්ධනය සම්බන්ධීකරණය කරන ලාභ නොලබන සංවිධානයකි

इलेक्ट्रॉनिकी एवं
सूचना प्रौद्योगिकी मंत्रालय
MINISTRY OF
ELECTRONICS AND
INFORMATION TECHNOLOGY
सत्यमेव जयते

It has expanded the character set to over 143,000 characters covering 154 modern and historic scripts, as well as symbols, emojis, and other graphical characters and even musical notation.
එය මේ වන විට නවීන සහ ඓතිහාසික අක්ෂර වර්ග 154ක් මෙන්ම සංකේත, ඉමෝජි සහ වෙනත් චිත්‍රක අක්ෂර සහ සංගීත අංකනය පවා ආවරණය වන පරිදි අක්ෂර කට්ටලය අක්ෂර 143,000කට වඩා පුළුල් කර ඇත.

Unicode can theoretically handle over a million code points, futureproofing it for generations to come.
යුනිකෝඩ් හට න්‍යායාත්මකව අක්ෂර මිලියනයකට වඩා හැසිරවිය හැකි අතර, එය ඉදිරි පරම්පරාවන් සඳහා අනාගතයේදී භාවිතය තහවුරු කරයි.

Unicode includes ASCII as a subset.
යුනිකෝඩ්හි උප කුලකයක් ලෙස ASCII ඇතුළත් වේ.



NOTE:
EBCDIC is not a subset of Unicode.
යුනිකෝඩ්හි EBCDIC උප කුලකයක් නොවේ.

Unicode supports several encoding forms.
යුනිකෝඩ් කේතන ආකාර කිහිපයකට සහය දක්වයි.

1. UTF-8
UTF-8 is a variable-length encoding scheme that can represent every Unicode character using one to four bytes.
UTF-8 යනු බයිටයක් හෝ හතරක් භාවිතා කරමින් සෑම යුනිකෝඩ් අක්ෂරයක්ම නියෝජනය කළ හැකි විචල්‍ය-දිග කේතන ක්‍රමයකි.

Commonly used on the web, in databases, and for file storage due to its efficiency and compatibility.
එහි කාර්යක්ෂමතාව සහ ගැළපුම හේතුවෙන් අන්තර්ජාලයේ, දත්ත සමුදායේ සහ ගොනු ගබඩා කිරීම සඳහා බහුලව භාවිතා වේ.

2. UTF-16
Another variable-length encoding scheme that uses either two or four bytes for each character is known as UTF-16.
එක් එක් අක්ෂර සඳහා බයිට දෙකක් හෝ හතරක් භාවිතා කරන තවත් විචල්‍ය-දිග කේතීකරණ ක්‍රමයක් ලෙස UTF-16 හඳුන්වයි.

Widely used in environments like Windows and Java, and suitable for applications where memory efficiency is less critical than processing speed.
වින්ඩෝස් සහ ජාවා වැනි පරිසරවල බහුලව භාවිතා වන අතර, සැකසුම් වේගයට වඩා මතක කාර්යක්ෂමතාව අඩු තීරණාත්මක යෙදුම් සඳහා සුදුසු වේ.

3. UTF-32
UTF-32 uses a fixed-length encoding of four bytes for every character.
UTF-32 සෑම අක්ෂරයක් සඳහාම බයිට හතරක ස්ථාවර දිග කේතනයක් භාවිතා කරයි.

Simplifies character processing since every character has the same byte size, making it easier to calculate offsets.
සෑම අක්ෂරයකටම එකම බයිට ප්‍රමාණය ඇති බැවින් අක්ෂර සැකසීම සරල කරයි, එය ඕෆ්සෙට් ගණනය කිරීම පහසු කරයි.

According to the syllabus, each character is represented by 16 bits.
විෂය නිර්දේශයට අනුව සෑම සංකේතයක්ම බිටි 16 කින් නිරූපණය කරයි

Number of maximum characters that can be represented by 16 bits combinations.
බිටු 16 ක සංයෝජන භාවිතා කර නිරූපණය කලහැකි උපරිම සංකේත ප්‍රමාණය

$= 2^{16} = 65536$

Consider the word "Hello".
"Hello" යන වචනය සලකා බලන්න.

ASCII

Each character uses 1 byte, so the word "Hello" would take 5 bytes.
සෑම අනුලක්ෂණයක්ම බයිටයක් භාවිතා කරයි, එබැවින් "Hello" යන වචනය බයිට 5ක් ගනී.

UTF-32
Each character uses 4 bytes, so "Hello" would take 20 bytes.
සෑම අනුලක්ෂණයක්ම බයිට 4ක් භාවිතා කරයි, එබැවින් "Hello" බයිට 20ක් ගනී.

Thus, Unicode encodings that use more bits per character require more storage and transmission time than ASCII for the same text.
මේ අනුව, එක් අනුලක්ෂණයට වැඩි බිටු භාවිතා කරන යුනිකෝඩ් කේතකරණය එකම පාඨ සඳහා ASCII වඩා වැඩි ගබඩා ඉඩක් සහ සම්ප්‍රේෂණ කාලය අවශ්‍ය වේ.

Comparison between several Unicode forms
යුනිකෝඩ් ආකෘති කිහිපයක් අතර සංසන්දනය

Encoding form **කේතන ආකෘතිය**	**Byte length** **බයිට දිග**	**Maximum Characters** **උපරිම අනුලක්ෂණ ගණන**	**Characteristics** **ලක්ෂණ**	**Common uses** **පොදු භාවිතයන්**
UTF-8	1 to 4 bytes	1,112,064	Can change size, works with ASCII, popular online ප්‍රමාණය වෙනස් කළ හැකිය, ASCII සමඟ ක්‍රියා කරයි, අන්තර්ජාලයේ ජනප්‍රිය වේ	Web pages, databases, email වෙබ් පිටු, දත්ත සමුදායන්, ඊමේල්
UTF-16	2 or 4 bytes	1,112,064	Most characters use 2 bytes; some use 4 bytes බොහෝ අක්ෂර බයිට 2ක් භාවිතා කරයි; සමහරක් බයිට 4ක් පාවිච්චි කරයි	Windows, Java applications වින්ඩෝස්, ජාවා වැනි යෙදුම්
UTF-32	4 bytes	1,112,064	Same size for all characters, easy to calculate සියලුම අක්ෂර සඳහා එකම ප්‍රමාණය, ගණනය කිරීමට පහසුය	Text processing, internal use පාඨ සැකසීම, අභ්‍යන්තර නිරූපණය

	040	041	042	043	044	045	046	047	048	049	04A	04B	04C	04D	04E	04F
0	Ѐ 0400	А 0410	Р 0420	а 0430	р 0440	ѐ 0450	Ѡ 0460	Ѱ 0470	Ҁ 0480	Ґ 0490	Ҡ 04A0	Ұ 04B0	Ӏ 04C0	Ӑ 04D0	Ӡ 04E0	Ӱ 04F0
1	Ё 0401	Б 0411	С 0421	б 0431	с 0441	ё 0451	ѡ 0461	ѱ 0471	ҁ 0481	ґ 0491	ҡ 04A1	ұ 04B1	Ӂ 04C1	ӑ 04D1	ӡ 04E1	ӱ 04F1
2	Ђ 0402	В 0412	Т 0422	в 0432	т 0442	ђ 0452	Ѣ 0462	Ѳ 0472	҂ 0482	Ғ 0492	Ң 04A2	Ҳ 04B2	ӂ 04C2	Ӓ 04D2	Ӣ 04E2	Ӳ 04F2
3	Ѓ 0403	Г 0413	У 0423	г 0433	у 0443	ѓ 0453	ѣ 0463	ѳ 0473	◌҃ 0483	ғ 0493	ң 04A3	ҳ 04B3	Ӄ 04C3	ӓ 04D3	ӣ 04E3	ӳ 04F3
4	Є 0404	Д 0414	Ф 0424	д 0434	ф 0444	є 0454	Ѥ 0464	Ѵ 0474	◌҄ 0484	Ҕ 0494	Ҥ 04A4	Ҵ 04B4	ӄ 04C4	Ӕ 04D4	Ӥ 04E4	Ӵ 04F4
5	Ѕ 0405	Е 0415	Х 0425	е 0435	х 0445	ѕ 0455	ѥ 0465	ѵ 0475	◌҅ 0485	ҕ 0495	ҥ 04A5	ҵ 04B5	Ӆ 04C5	ӕ 04D5	ӥ 04E5	ӵ 04F5
6	І 0406	Ж 0416	Ц 0426	ж 0436	ц 0446	і 0456	Ѧ 0466	Ѷ 0476	◌҆ 0486	Җ 0496	Ҧ 04A6	Ҷ 04B6	ӆ 04C6	Ӗ 04D6	Ӧ 04E6	Ӷ 04F6
7	Ї 0407	З 0417	Ч 0427	з 0437	ч 0447	ї 0457	ѧ 0467	ѷ 0477	◌҇ 0487	җ 0497	ҧ 04A7	ҷ 04B7	Ӈ 04C7	ӗ 04D7	ӧ 04E7	ӷ 04F7
8	Ј 0408	И 0418	Ш 0428	и 0438	ш 0448	ј 0458	Ѩ 0468	Ѹ 0478	◌҈ 0488	Ҙ 0498	Ҩ 04A8	Ҹ 04B8	ӈ 04C8	Ә 04D8	Ө 04E8	Ӹ 04F8
9	Љ 0409	Й 0419	Щ 0429	й 0439	щ 0449	љ 0459	ѩ 0469	ѹ 0479	◌҉ 0489	ҙ 0499	ҩ 04A9	ҹ 04B9	Ӊ 04C9	ә 04D9	ө 04E9	ӹ 04F9
A	Њ 040A	К 041A	Ъ 042A	к 043A	ъ 044A	њ 045A	Ѫ 046A	Ѻ 047A	Ҋ 048A	Қ 049A	Ҫ 04AA	Һ 04BA	ӊ 04CA	Ӛ 04DA	Ӫ 04EA	Ӻ 04FA
B	Ћ 040B	Л 041B	Ы 042B	л 043B	ы 044B	ћ 045B	ѫ 046B	ѻ 047B	ҋ 048B	қ 049B	ҫ 04AB	һ 04BB	Ӌ 04CB	ӛ 04DB	ӫ 04EB	ӻ 04FB
C	Ќ 040C	М 041C	Ь 042C	м 043C	ь 044C	ќ 045C	Ѭ 046C	Ѽ 047C	Ҍ 048C	Ҝ 049C	Ҭ 04AC	Ҽ 04BC	ӌ 04CC	Ӝ 04DC	Ӭ 04EC	Ӽ 04FC
D	Ѝ 040D	Н 041D	Э 042D	н 043D	э 044D	ѝ 045D	ѭ 046D	ѽ 047D	ҍ 048D	ҝ 049D	ҭ 04AD	ҽ 04BD	Ӎ 04CD	ӝ 04DD	ӭ 04ED	ӽ 04FD
E	Ў 040E	О 041E	Ю 042E	о 043E	ю 044E	ў 045E	Ѯ 046E	Ѿ 047E	Ҏ 048E	Ҟ 049E	Ү 04AE	Ҿ 04BE	ӎ 04CE	Ӟ 04DE	Ӯ 04EE	Ӿ 04FE
F	Џ 040F	П 041F	Я 042F	п 043F	я 044F	џ 045F	ѯ 046F	ѿ 047F	ҏ 048F	ҟ 049F	ү 04AF	ҿ 04BF	ӏ 04CF	ӟ 04DF	ӯ 04EF	ӿ 04FF

	304	305	306	307	308	309
0		ぐ 3050	だ 3060	ば 3070	む 3080	ゐ 3090
1	ぁ 3041	け 3051	ち 3061	ぱ 3071	め 3081	ゑ 3091
2	あ 3042	げ 3052	ぢ 3062	ひ 3072	も 3082	を 3092
3	ぃ 3043	こ 3053	っ 3063	び 3073	ゃ 3083	ん 3093
4	い 3044	ご 3054	つ 3064	ぴ 3074	や 3084	ゔ 3094
5	ぅ 3045	さ 3055	づ 3065	ふ 3075	ゅ 3085	ゕ 3095
6	う 3046	ざ 3056	て 3066	ぶ 3076	ゆ 3086	ゖ 3096
7	ぇ 3047	し 3057	で 3067	ぷ 3077	ょ 3087	
8	え 3048	じ 3058	と 3068	へ 3078	よ 3088	
9	ぉ 3049	す 3059	ど 3069	べ 3079	ら 3089	◌゙ 3099
A	お 304A	ず 305A	な 306A	ぺ 307A	り 308A	◌゚ 309A
B	か 304B	せ 305B	に 306B	ほ 307B	る 308B	゛ 309B
C	が 304C	ぜ 305C	ぬ 306C	ぼ 307C	れ 308C	゜ 309C
D	き 304D	そ 305D	ね 306D	ぽ 307D	ろ 308D	ゝ 309D
E	ぎ 304E	ぞ 305E	の 306E	ま 307E	ゎ 308E	ゞ 309E
F	く 304F	た 305F	は 306F	み 307F	わ 308F	ゟ 309F

	0D8	0D9	0DA	0DB	0DC	0DD	0DE	0DF
0		ඐ 0D90	ච 0DA0	ධ 0DB0	ව 0DC0	◌ැ 0DD0		
1	◌ඁ 0D81	එ 0D91	ඡ 0DA1	න 0DB1	ශ 0DC1	◌ෑ 0DD1		
2	◌ං 0D82	ඒ 0D92	ජ 0DA2		ෂ 0DC2	◌ි 0DD2		◌ෲ 0DF2
3	◌ඃ 0D83	ඓ 0D93	ඣ 0DA3	ඳ 0DB3	ස 0DC3	◌ී 0DD3		◌ෳ 0DF3
4		ඔ 0D94	ඤ 0DA4	ප 0DB4	හ 0DC4	◌ු 0DD4		෴ 0DF4
5	අ 0D85	ඕ 0D95	ඥ 0DA5	ඵ 0DB5	ළ 0DC5			
6	ආ 0D86	ඖ 0D96	ඦ 0DA6	බ 0DB6	ෆ 0DC6	◌ූ 0DD6	෦ 0DE6	
7	ඇ 0D87		ට 0DA7	භ 0DB7			෧ 0DE7	
8	ඈ 0D88		ඨ 0DA8	ම 0DB8		◌ෘ 0DD8	෨ 0DE8	
9	ඉ 0D89		ඩ 0DA9	ඹ 0DB9		◌ෙ 0DD9	෩ 0DE9	
A	ඊ 0D8A	ක 0D9A	ඪ 0DAA	ය 0DBA	◌් 0DCA	◌ේ 0DDA	෪ 0DEA	
B	උ 0D8B	ඛ 0D9B	ණ 0DAB	ර 0DBB		◌ෛ 0DDB	෫ 0DEB	
C	ඌ 0D8C	ග 0D9C	ඬ 0DAC			◌ො 0DDC	෬ 0DEC	
D	ඍ 0D8D	ඝ 0D9D	ත 0DAD	ල 0DBD		◌ෝ 0DDD	෭ 0DED	
E	ඎ 0D8E	ඞ 0D9E	ථ 0DAE			◌ෞ 0DDE	෮ 0DEE	
F	ඏ 0D8F	ඟ 0D9F	ද 0DAF		◌ා 0DCF	◌ෟ 0DDF	෯ 0DEF	

	000	001	002	003	004	005	006	007
0	NUL 0000	DLE 0010	SP 0020	0 0030	@ 0040	P 0050	` 0060	p 0070
1	SOH 0001	DC1 0011	! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071
2	STX 0002	DC2 0012	" 0022	2 0032	B 0042	R 0052	b 0062	r 0072
3	ETX 0003	DC3 0013	# 0023	3 0033	C 0043	S 0053	c 0063	s 0073
4	EOT 0004	DC4 0014	$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074
5	ENQ 0005	NAK 0015	% 0025	5 0035	E 0045	U 0055	e 0065	u 0075
6	ACK 0006	SYN 0016	& 0026	6 0036	F 0046	V 0056	f 0066	v 0076
7	BEL 0007	ETB 0017	' 0027	7 0037	G 0047	W 0057	g 0067	w 0077
8	BS 0008	CAN 0018	(0028	8 0038	H 0048	X 0058	h 0068	x 0078
9	HT 0009	EM 0019	) 0029	9 0039	I 0049	Y 0059	i 0069	y 0079
A	LF 000A	SUB 001A	* 002A	: 003A	J 004A	Z 005A	j 006A	z 007A
B	VT 000B	ESC 001B	+ 002B	; 003B	K 004B	[005B	k 006B	{ 007B
C	FF 000C	FS 001C	, 002C	< 003C	L 004C	\ 005C	l 006C	| 007C
D	CR 000D	GS 001D	- 002D	= 003D	M 004D	] 005D	m 006D	} 007D
E	SO 000E	RS 001E	. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E
F	SI 000F	US 001F	/ 002F	? 003F	O 004F	_ 005F	o 006F	DEL 007F

	090	091	092	093	094	095	096	097
0	◌ऀ 0900	ऐ 0910	ठ 0920	र 0930	◌ी 0940	ॐ 0950	ॠ 0960	॰ 0970
1	◌ँ 0901	ऑ 0911	ड 0921	ऱ 0931	◌ु 0941	◌॑ 0951	ॡ 0961	ॱ 0971
2	◌ं 0902	ऒ 0912	ढ 0922	ल 0932	◌ू 0942	◌॒ 0952	◌ॢ 0962	ॲ 0972
3	◌ः 0903	ओ 0913	ण 0923	ळ 0933	◌ृ 0943	◌॓ 0953	◌ॣ 0963	ॳ 0973
4	ऄ 0904	औ 0914	त 0924	ऴ 0934	◌ॄ 0944	◌॔ 0954	। 0964	ॴ 0974
5	अ 0905	क 0915	थ 0925	व 0935	◌ॅ 0945	◌ॕ 0955	॥ 0965	ॵ 0975
6	आ 0906	ख 0916	द 0926	श 0936	◌ॆ 0946	◌ॖ 0956	० 0966	ॶ 0976
7	इ 0907	ग 0917	ध 0927	ष 0937	◌े 0947	◌ॗ 0957	१ 0967	ॷ 0977
8	ई 0908	घ 0918	न 0928	स 0938	◌ै 0948	क़ 0958	२ 0968	ॸ 0978
9	उ 0909	ङ 0919	ऩ 0929	ह 0939	◌ॉ 0949	ख़ 0959	३ 0969	ॹ 0979
A	ऊ 090A	च 091A	प 092A	◌ऺ 093A	◌ॊ 094A	ग़ 095A	४ 096A	ॺ 097A
B	ऋ 090B	छ 091B	फ 092B	◌ऻ 093B	◌ो 094B	ज़ 095B	५ 096B	ॻ 097B
C	ऌ 090C	ज 091C	ब 092C	◌़ 093C	◌ौ 094C	ड़ 095C	६ 096C	ॼ 097C
D	ऍ 090D	झ 091D	भ 092D	ऽ 093D	◌् 094D	ढ़ 095D	७ 096D	ॽ 097D
E	ऎ 090E	ञ 091E	म 092E	◌ा 093E	◌ॎ 094E	फ़ 095E	८ 096E	ॾ 097E
F	ए 090F	ट 091F	य 092F	◌ि 093F	◌ॏ 094F	य़ 095F	९ 096F	ॿ 097F

	0B8	0B9	0BA	0BB	0BC	0BD	0BE	0BF
0		ஐ 0B90		ர 0BB0	◌ீ 0BC0	ௐ 0BD0		௰ 0BF0
1				ற 0BB1	◌ு 0BC1			௱ 0BF1
2	◌ஂ 0B82	ஒ 0B92		ல 0BB2	◌ூ 0BC2			௲ 0BF2
3	ஃ 0B83	ஓ 0B93	ண 0BA3	ள 0BB3				௳ 0BF3
4		ஔ 0B94	த 0BA4	ழ 0BB4				௴ 0BF4
5	அ 0B85	க 0B95		வ 0BB5				௵ 0BF5
6	ஆ 0B86			ஶ 0BB6	◌ெ 0BC6		௦ 0BE6	௶ 0BF6
7	இ 0B87			ஷ 0BB7	◌ே 0BC7	◌ௗ 0BD7	௧ 0BE7	௷ 0BF7
8	ஈ 0B88		ந 0BA8	ஸ 0BB8	◌ை 0BC8		௨ 0BE8	௸ 0BF8
9	உ 0B89	ங 0B99	ன 0BA9	ஹ 0BB9			௩ 0BE9	௹ 0BF9
A	ஊ 0B8A	ச 0B9A	ப 0BAA		◌ொ 0BCA		௪ 0BEA	௺ 0BFA
B					◌ோ 0BCB		௫ 0BEB	
C		ஜ 0B9C			◌ௌ 0BCC		௬ 0BEC	
D					◌் 0BCD		௭ 0BED	
E	எ 0B8E	ஞ 0B9E	ம 0BAE	◌ா 0BBE			௮ 0BEE	
F	ஏ 0B8F	ட 0B9F	ய 0BAF	◌ி 0BBF			௯ 0BEF	

Advantages and Disadvantages of Unicode coding system
යුනිකෝඩ් කේතකරණ පද්ධතියේ වාසි සහ අවාසි

Aspect	**Advantages** **වාසි**	**Disadvantages** **අවාසි**
Language Support **භාෂා සහාය**	Works with most languages බොහෝ භාෂා සමඟ ක්‍රියා කරයි.	Some rare languages and symbols are missing ඇතැම් දුර්ලභ භාෂා සහ සංකේත අන්තර්ගත කර ඇත.
Consistency **අනුකූලතාව**	Same character looks the same everywhere එකම අනුලක්ෂණ සැමතැනම එක අයුරින් පවතී.	Some old systems may not work with Unicode සමහර පැරණි පද්ධති යුනිකෝඩ් සමඟ ක්‍රියා නොකරනු ඇත.
Special Characters **විශේෂ අනුලක්ෂණ**	Has symbols, emojis, and math characters සංකේත, ඉමෝජි සහ ගණිත අක්ෂර ඇත.	Needs the right font type to display properly නිවැරදිව පෙන්වීමට නිවැරදි අකුරු වර්ගය අවශ්‍ය වේ.
Encoding Flexibility **කේතීකරණ නම්‍යශීලී බව**	UTF-8, UTF-16, and UTF-32 use different sizes for storage UTF-8, UTF-16 සහ UTF-32 ගබඩා කිරීම සඳහා විවිධ ප්‍රමාණ භාවිතා කරයි.	Complicated to work with because sizes vary ප්‍රමාණ වෙනස් නිසා ක්‍රියාත්මක වීම සංකීර්ණ වේ.
ASCII Compatibility **ASCII අනුකූලතාව**	Matches older ASCII codes for easy use පහසු භාවිතය සඳහා පැරණි ASCII කේත ගැලපේ.	Uses more space and memory for non-ASCII text ASCII නොවන පාඨ සඳහා වැඩි ඉඩක් සහ මතකයක් භාවිත කරයි.
Web and Software Standard **වෙබ් සහ මෘදුකාංග සම්මතය**	Used widely on websites and in software වෙබ් අඩවි වල සහ මෘදුකාංග වල බහුලව භාවිතා වේ.	Older programs may not fully support it පැරණි වැඩසටහන් එයට සම්පූර්ණයෙන්ම සහාය නොදක්වයි.

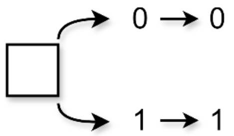
Feature **ලක්ෂණ**	**BCD**	**ASCII**	**EBCDIC**	**Unicode**
Purpose **අරමුණ**	Represents decimal numbers in binary form ද්විමය ආකාරයෙන් දශම සංඛ්‍යා නියෝජනය කරයි	Represents basic English characters and symbols මූලික ඉංග්‍රීසි අනුලක්ෂණ සහ සංකේත නියෝජනය කරයි	Represents characters mainly on IBM mainframes ප්‍රධාන වශයෙන් IBM ප්‍රධාන රාමු වල අනුලක්ෂණ නියෝජනය කරයි	Represents characters from all languages and symbols globally ගෝලීය වශයෙන් සියලුම භාෂා සහ සංකේත වලින් අනුලක්ෂණ නියෝජනය කරයි
Bit Size **බිටු ප්‍රමාණය**	4 bits per decimal digit දශම ඉලක්කමකට බිටු 4ක්	7 bits (often extended to a full byte) බිටු 7 (බොහෝ විට පූර්ණ බයිටයක් දක්වා දිර්ඝ වේ)	8 bits (1 byte) බිටු 8 (බයිට 1ක්)	Variable / විචල්‍ය (UTF-8: 1-4 bytes, UTF-16: 2 bytes, UTF-32: 4 bytes)
Character Support **අනුලක්ෂණ සහාය**	Supports only numbers 0-9 සංඛ්‍යා 0-9 සඳහා පමණක් සහය දක්වයි	Supports 128 characters (English letters, numbers, symbols) අනුලක්ෂණ 128 සඳහා සහය දක්වයි (ඉංග්‍රීසි අකුරු, අංක, සංකේත)	Supports 256 characters, mainly English and symbols අනුලක්ෂණ 256 ප්‍රධාන වශයෙන් ඉංග්‍රීසි සහ සංකේත සඳහා සහය දක්වයි	Supports thousands of characters and symbols (languages, emojis, etc.) අනුලක්ෂණ සහ සංකේත දහස් ගණනකට සහය දක්වයි (භාෂා, ඉමෝජි, ආදිය)
Compatibility **ගැළපුම**	Not compatible with ASCII or Unicode ASCII හෝ Unicode සමඟ නොගැළපේ	Widely compatible in older systems පැරණි පද්ධතිවලට පුළුල් ලෙස අනුකූල වේ	Not compatible with ASCII or Unicode ASCII හෝ Unicode සමඟ නොගැළපේ	Includes ASCII compatibility ASCII අනුකූලතාව ඇතුළත් වේ
Usage **භාවිතය**	Calculations and digital displays ගණනය කිරීම් සහ අංකිත සංදර්ශක	Basic text files, legacy systems මූලික පාඨ ගොනු, උරුම පද්ධති	IBM mainframes, older financial and business systems IBM ප්‍රධාන රාමු වල, පැරණි මූල්‍ය සහ ව්‍යාපාර පද්ධති	Takes more space and time to transmit සම්ප්‍රේෂණය කිරීමට වැඩි ඉඩක් සහ කාලයක් ගතවේ

Representing negative numbers in computers
පරිගණකවල සෘණ සංඛ්‍යා නිරූපණය කිරීම

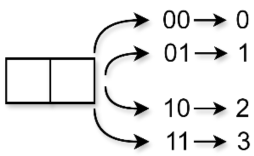
Unsigned magnitude representation
ලකුණු රහිත සංඛ්‍යා නිරූපණය

Here, a binary number is always considered to be a positive number. Negative numbers cannot be represented using this method.
මෙහිදී ද්වීමය සංඛ්‍යාවක් සෑමවිටම ධන සංඛ්‍යාවක් ලෙස සලකයි. මෙම ක්‍රමය මගින් සෘණ සංඛ්‍යා නිරූපණය කිරීම සිදුකළ නොහැක.

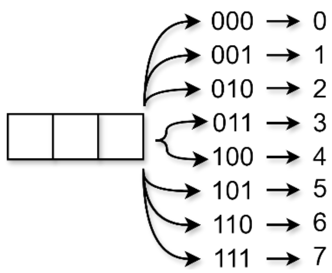
1 bit - 0 and 1



2 bits - 00 to 11 (3_{10})



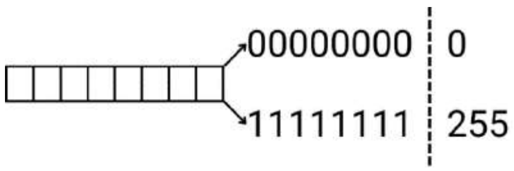
3 bits - 000 to 111 (7_{10})

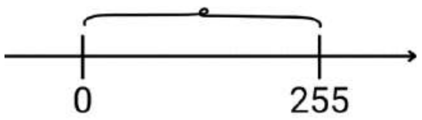


4 bits - 0000 to 1111 (15_{10})
5 bits - 00000 to 11111 (31_{10})
6 bits - 000000 to 111111 (63_{10})
7 bits - 0000000 to 1111111 (127_{10})

Maximum range of numbers which can be represented using n bits බිටු n ගණනක් යොදාගනිමින් නිරූපණය කළ හැකි උපරිම සංඛ්‍යා පරාසය	0 to (2^N-1)

8 bits - 00000000 to 11111111 (255_{10})





Examples:
29 = 00011101_2
90 =01011010_2
220 = 11011100_2

NOTE:
Any number of bits can be used while using negative number representation methods but according to the syllabus 8 bits must be used.
සෘණ සංඛ්‍යා නිරූපණය කිරීමේ ක්‍රම භාවිතා කිරීමේදී ඕනෑම බිටු ප්‍රමාණයක් භාවිතා කළ හැකි වුවද විෂය නිර්දේශයට අනුව අනිවාර්යයෙන් බිටු 8ක් භාවිතා කළ යුතුය.

Signed magnitude representation.
ලකුණු සහිත සංඛ්‍යා නිරූපණය

This is a straightforward way to represent positive and negative binary numbers.
මෙය ධන සහ ඍණ ද්විමය සංඛ්‍යා නිරූපනය කිරීමට සරල ක්‍රමයකි.

The leftmost bit of a number (MSB or the Most Significant Bit) represents the sign of the number.
සංඛ්‍යාවක වම්පසම ඇති බිටුව (MSB හෝ වැඩිම වෙසෙසි බිටුව) එම සංඛ්‍යාවේ ලකුණ නියෝජනය කරයි.

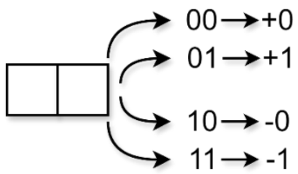
'0' indicates a positive number.
'0' ධන සංඛ්‍යාවක් නිරූපණය කරයි.

'1' indicates a negative number.
'1' ඍණ සංඛ්‍යාවක් නිරූපණය කරයි.

The remaining bits represent the absolute value (or magnitude) of the number.
ඉතිරි බිටු සංඛ්‍යාවේ නිරපේක්ෂ අගය (හෝ විශාලත්වය) නිරූපණය කරයි.

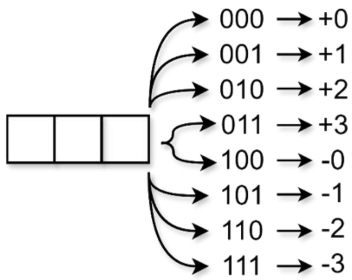
NOTE:
Single bit cannot exist in a signed magnitude representation. There should be at least 2.
ලකුණු සහිත සංඛ්‍යා නිරූපණයේදී තනි බිටුවක් පමණක් පැවතිය නොහැක. අවම වශයෙන් 2 ක් වත් තිබිය යුතුය.

When we take 2-bit numbers,
බිටු 2 හි සංඛ්‍යා සැලකීමේදී,



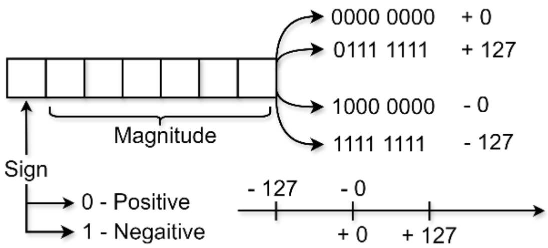
00	is normally 0	but now +0
01	is normally 1	but now +1
10	is normally 2	but now -0
11	is normally 3	but now -1

When we take 3-bit numbers,
බිටු 3 හි සංඛ්‍යා සැලකීමේදී,



000	is normally 0	but now +0
001	is normally 1	but now +1
010	is normally 2	but now +2
011	is normally 3	but now +3
100	is normally 4	but now -0
101	is normally 5	but now -1
110	is normally 6	but now -2
111	is normally 7	but now -3

Usually, we take 8 bits to represent the number as the default.
සාමාන්‍යයෙන් සංඛ්‍යාව සම්මත ලෙස නිරූපණය කිරීම සඳහා බිටු 8ක් භාවිතා කරයි.



Using this method, we can represent both negative and positive values using 8 bits ranging from -127 up to +127 instead of representing positive numbers ranging from 0 to 255.
මෙම ක්‍රමයෙන් බිටු 8ක් භාවිතා කර 0 සිට 255 දක්වා වූ ධන සංඛ්‍යා නියෝජනය කරනවා වෙනුවට -127 සිට +127 දක්වා වූ ඍණ සහ ධන අගයන් දෙකම නිරූපණය කළ හැක.

Example:
in 8-bit sign-magnitude

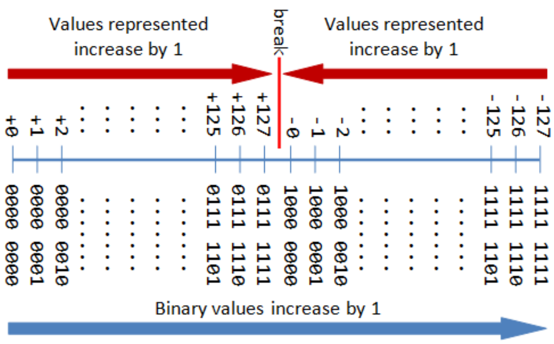
Positive 5 is	00000101.
Negative 5 is	10000101.

If we take less than or more than 8 bits the range will differ as follows
බිටු 8කට වඩා අඩුවෙන් හෝ වැඩි වශයෙන් ලබාගත් විට පරාසය පහත පරිදි වෙනස් වේ.

1 bit	incompatible
2 bits	-1 to +1
3 bits	-3 to +3
4 bits	-7 to +7
5 bits	-15 to +15
6 bits	-31 to +31
7 bits	-63 to +63
8 bits	-127 to +127
9 bits	-255 to +255
10 bits	-511 to +511

Examples:

+43	=	0	0101011
-58	=	1	0111010
+99	=	0	1100011
-123	=	1	1111011
		Sign bit	



Sign-Magnitude Representation

Advantages of Signed magnitude representation.
ලකුණු සහිත සංඛ්‍යා නිරූපණයෙහි වාසි

This resembles human use of positive and negative decimals, separating the sign from the magnitude for easier understanding.
මෙය ධන සහ සෘණ දශමවල මානව භාවිතයට සමාන වන අතර, පහසුවෙන් අවබෝධ කර ගැනීම සඳහා ලකුණ විශාලත්වයෙන් වෙන් කරයි.

Hardware design is simpler. Converting between positive and negative involves flipping the sign bit.
දෘඩාංග නිර්මාණය සරලයි. ධණ සහ සෘණ අතර පරිවර්තනය කිරීම සංඥා බිටු පෙරලීම ඇතුළත් වේ.

Disadvantages of Signed magnitude representation.
ලකුණු සහිත සංඛ්‍යා නිරූපණයෙහි අවාසි

Two zeros are represented, one positive and one negative.
ශුන්‍ය දෙකක් නියෝජනය වේ, එකක් ධන සහ අනේක සෘණ.

This leads to complications in hardware and software and the number of representable values is one less considering other representations of the same number of bits.
මෙය දෘඩාංග සහ මෘදුකාංග වල සංකූලතා ඇති කරන අතර එකම බිටු සංඛ්‍යාව භාවිතා කරන අනෙකුත් නිරූපණ සලකා බැලීමේදී මෙමගින් නිරූපණය කළ හැකි අගයන් සංඛ්‍යාව එකක් අඩු වේ.

Arithmetic operations like addition and subtraction become complex, as incorrect results may occur without separately evaluating the sign and magnitude.
ලකුණ සහ විශාලත්වය වෙන වෙනම ඇගයීමකින් තොරව වැරදි ප්‍රතිඵල ඇති විය හැකි බැවින්, එකතු කිරීම සහ අඩු කිරීම වැනි අංක ගණිත ක්‍රියා සංකීර්ණ වේ.

Complementary arithmetic
අනුපූරක ගණිතය

Complementary arithmetic is a method of subtraction by addition.
අනුපූරක ගණිතය යනු එකතු කිරීම අනුසාරයෙන් අඩු කිරීම සිදු කරන ක්‍රමයකි.

For that, we should understand the concept of positive and negative.
ඒ සඳහා ධන සෘණ සංකල්පය පිළිබඳව අවබෝධයක් ලබාගෙන සිටිය යුතුය.

+5	+	+2	=	$\boxed{+7}$
+5	+	-2	=	$\boxed{+3}$
-5	+	+2	=	$\boxed{-3}$
-5	+	-2	=	$\boxed{-7}$
+5	-	+2	=	$\boxed{+3}$
+5	-	-2	=	$\boxed{+7}$
-5	-	+2	=	$\boxed{-7}$
-5	-	-2	=	$\boxed{-3}$

9's complement (Not in syllabus)

Here, let's use base ten numbers with 4 digits for the calculation.
මෙහිදී ගණනය කිරීම සඳහා ඉලක්කම් 4ක් සහිත දහයේ පාදයේ සංඛ්‍යා භාවිතා කරමු.

The second number should be smaller than the first one used for the calculation.
මෙහිදී ගණනය කිරීම සඳහා යොදාගන්නා පළමු සංඛ්‍යාවට වඩා දෙවන සංඛ්‍යාව කුඩා විය යුතුය.

Example:

	10	10	10	
4 ~~5~~	3 ~~4~~	1 ~~2~~	3_{10}	n1
- 0	9	8	7_{10}	n2
4	4	3	6_{10}	

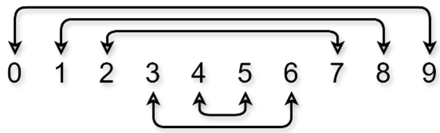
9	9	9	9_{10}	
- 0	9	8	7_{10}	n2
9	0	1	2_{10}	n3
				[9's com]

	5	4	2	3_{10}	n1
+	9	0	1	2_{10}	n3
1	4	4	3	5	
			+	1	
	4	4	3	6_{10}	

overflow

Here in the last step, 1 is added only to get the respective answer.
මෙහි අවසාන පියවරේදී අදාළ පිළිතුර ලබාගැනීම සඳහා පමණක් 1ක් එකතු කරනු ලැබේ.

But here, a subtraction is used to obtain n3. As an alternative to this, the number replacement method can be used.
නමුත් මෙහි n3 ලබාගන්නා විට අඩු කිරීමක් භාවිතා කර ඇත. මෙයට විකල්පයක් ලෙස සංඛ්‍යා ප්‍රතිස්ථාපන ක්‍රමය භාවිතා කළ හැකිය.



10's complement (Not in syllabus)

Here let's use base ten numbers with 4 digits for calculation.
මෙහිදී ගණනය කිරීම සඳහා ඉලක්කම් 4ක් සහිත දහයේ පාදයේ සංඛ්‍යා භාවිතා කරමු.

Example:

	10	10	10	
4 ~~5~~	3 ~~4~~	1 ~~2~~	3_{10}	n1
- 0	9	8	7_{10}	n2
4	4	3	6_{10}	

9	9	9	9_{10}	
- 0	9	8	7_{10}	n2
9	0	1	2_{10}	n3 [9's com]
		+	1	
9	0	1	3	n4 [10's com]

	5	4	2	3_{10}	n1
+	9	0	1	3_{10}	n4
1	4	4	3	6_{10}	

1 → overflow

7's complement (Not in syllabus)

Here let's use base eight numbers with 5 digits for calculation.

මෙහිදී ගණනය කිරීම සඳහා ඉලක්කම් 5ක් සහිත අපේ පාදයේ සංඛ්‍යා භාවිතා කරමු.

Example:

	7 ~~8~~	8	8	8	
2 ~~3~~	0	3 ~~4~~	1 ~~2~~	1_8	n1
- 0	5	7	6	2_8	n2
2	2	4	3	7_8	

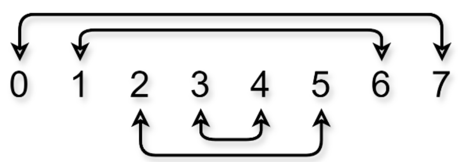
7	7	7	7	7_8	
- 0	5	7	6	2_8	n2
7	2	0	1	5_8	n3 [7's com]

	3	0	4	2	1	n1
+	7	2	0	1	5	n3
1	2	2	4	3	6	
				+	1	
	2	2	4	3	7_7	

1 → overflow

But here a subtraction is used to obtain n3. As an alternative to this, the number replacement method can be used.

නමුත් මෙහි n3 ලබාගන්නා විට අඩු කිරීමක් භාවිතා කර ඇත. මෙයට විකල්පයක් ලෙස සංඛ්‍යා ප්‍රතිස්ථාපන ක්‍රමය භාවිතා කළ හැකිය.



8's complement (Not in syllabus)

Here let's use base eight numbers with 5 digits for calculation.

මෙහිදී ගණනය කිරීම සඳහා ඉලක්කම් 5ක් සහිත අපේ පාදයේ සංඛ්‍යා භාවිතා කරමු.

Example:

	7 ~~8~~	8	8	8	
2 ~~3~~	0	3 ~~4~~	1 ~~2~~	1_8	n1
- 0	5	7	6	2_8	n2
2	2	4	3	7_8	

7	7	7	7	7_8	
- 0	5	7	6	2_8	n2
7	2	0	1	5_8	n3 [7's com]
			+	1	
7	2	0	1	6_8	n4 [8's com]

	3	0	4	2	1_8	n1
+	7	2	0	1	6_8	n4
1	2	2	4	3	7_8	

1 → overflow

1's complement
1හි අනුපූරකය

One's complement was a method used to represent positive and negative integers in early computing systems.
1හි අනුපූරකය යනු අතීතයේ පරිගණක පද්ධතිවල ධන සහ සෘණ නිඛිල නියෝජනය කිරීම සඳහා භාවිතා කළ ක්‍රමයකි.

Here the complement of a number is found by inverting all the bits.
මෙහිදී සංඛ්‍යාවක අනුපූරකය සොයාගනු ලබන්නේ සියලුම බිටු පෙරළීමෙනි.
(changing 1s to 0s and 0s to 1s).
(1 ඒවා 0 දක්වා සහ 0 ඒවා 1 දක්වා වෙනස් කිරීම).

It is used to represent the negative numbers of its original number.
එය මුල් සංඛ්‍යාවේ සෘණ සංඛ්‍යාව නිරූපණය කිරීමට භාවිතා කරයි.

Let's consider 2 bit One's complement.
බිටු 2 ක, එකෙහි අනුපූරකය සලකා බලමු.

00	is normally 0	but now +0
01	is normally 1	but now +1
10	is normally 2	but now -1
11	is normally 0	but now -0

Let's consider 3 bit One's complement.
බිටු 3, එකෙහි අනුපූරකය සලකා බලමු.

000	is normally 0	but now +0
001	is normally 1	but now +1
010	is normally 2	but now +2
011	is normally 3	but now +3
100	is normally 4	but now -3
101	is normally 5	but now -2
110	is normally 6	but now -1
111	is normally 7	but now -0

As the standard number of bits used to represent one's complement numbers in the syllabus is 8-bits, the binary number should be written in 8-bits whether it's either positive or negative.
විෂය නිර්දේශය තුළ එකෙහි අනුපූරක සංඛ්‍යා නියෝජනය කිරීමට භාවිතා කරන සම්මතය බිටු 8 සංඛ්‍යා බැවින් ධන හෝ සෘණ ඕනෑම ද්විමය සංඛ්‍යාවක් බිටු 8 කින් ලිවිය යුතුයි

Using this method, we can represent both negative and positive values using 8 bits ranging from -127 up to +127 instead of representing positive numbers ranging from 0 up to 255.
බිටු 8 ක් භාවිතා කරමින් මෙම ක්‍රමය යෙදූ විට අපට 0 සිට 255 දක්වා වූ ධන සංඛ්‍යා නියෝජනය කරනවා වෙනුවට -127 සිට +127 දක්වා වූ සෘණ සහ ධන අගයන් දෙකම නිරූපණය කළ හැකි

in 8-bit one's complement
Positive 5 is 00000101.
Negative 5 is 11111010.

If we take less than or more than 8 bits the range will differ as follows
බිටු 8කට වඩා අඩුවෙන් හෝ වැඩි වශයෙන් ලබාගත් විට පරාසය පහත පරිදි වෙනස් වේ

1 bit	incompatible
2 bits	-1 to +1
3 bits	-3 to +3
4 bits	-7 to +7
5 bits	-15 to +15
6 bits	-31 to +31
7 bits	-63 to +63
8 bits	-127 to +127
9 bits	-255 to +255
10 bits	-511 to + 511

This was used in some older computer systems for subtracting numbers by adding the one's complement of one number to another number.
මෙය සමහර පැරණි පරිගණක පද්ධතිවල එක් අංකයක අනුපූරකය තවත් අංකයකට එකතු කිරීමෙන් සංඛ්‍යා අඩු කිරීම සඳහා භාවිතා කරන ලදී.

This is also used in some error-detection systems.
මෙය සමහර දෝෂ හඳුනාගැනීමේ පද්ධති වලද භාවිතා වේ.

Example:
n1-n2 (n1>n2)

$$\begin{array}{rcccccccl}
 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1_2 \; \text{— n1} \\
- & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0_2 \; \text{— n2} \\
\hline
 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1_2 \\
\hline\hline
\end{array}$$

$$\begin{array}{rcccccccl}
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1_2 \\
- & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0_2 \; \text{— n2} \\
\hline
 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1_2 \; \text{— n3} \\
\hline\hline
\end{array}$$

[1's com]

NOT gate

$$\begin{array}{rcccccccl}
 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1_2 \; \text{— n1} \\
+ & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1_2 \; \text{— n3} \\
\hline
1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0_2 \\
 & & & & & & & + & 1 \\
\hline
 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\
\hline\hline
\end{array}$$

overflow

NOTE:
Here, the bit inversion can be done while getting the one's complement.
මෙහිදී එකෙහි අනුපූරකය ලබාගැනීමේදී බිටු ප්‍රතිලෝම කිරීම සිදු කළ හැකිය.

Here, “1” should be added at the end only if an overflow bit is present.
මෙහිදී අවසානයේ 1ක් එකතු කළ යුත්තේ ඉවත ගලායාමේ බිටුවක් පැවතුනහොත් පමණි

Example 02
n1-n2 (n1=n2)

$$\begin{array}{lll}
=5 - 5 \rightarrow 5 + (-5) & & \\
+5 & \rightarrow & 0\;0\;0\;0\;0\;1\;0\;1_2 \; \text{— n1} \\
-5 & \rightarrow & \text{Invert } +5 \\
 & & 1\;1\;1\;1\;1\;0\;1\;0_2 \; \text{— n2}
\end{array}$$

[1's com]

$$\begin{array}{rcccccccl}
 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1_2 \; \text{— n1} \\
+ & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0_2 \; \text{— n2} \\
\hline
 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1_2 \\
 & & & & & & & + & 1 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline\hline
\end{array}$$

overflow

Example 03
n1-n2 (n1<n2)

$$\begin{array}{lll}
=20 - 35 \rightarrow ? & & \\
= 20 + (-35) & & \\
+20 & \rightarrow & 0\;0\;0\;1\;0\;1\;0\;0_2 \; \text{— n1} \\
+35 & \rightarrow & 0\;0\;1\;0\;0\;0\;1\;1_2 \\
-35 & \rightarrow & \text{Invert } +35 \\
 & & 1\;1\;0\;1\;1\;1\;0\;0_2 \; \text{— n2}
\end{array}$$

[1's com]

$$\begin{array}{rcccccccl}
 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0_2 \; \text{— n1} \\
+ & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0_2 \; \text{— n2} \\
\hline
 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0_2 \\
\hline\hline
\end{array}$$

1 is not added because there is no discard bit here.
මෙහිදී ඉවතගලායාමේ බිටුවක් නොමැති නිසා 1ක් එකතු නොකරයි.

Advantages of 1’s complement
එකෙහි අනුපූරකයෙහි වාසි

1's complement is relatively easy to implement, requiring only bitwise inversion.
එකෙහි අනුපූරකය ක්‍රියාත්මක කිරීමට සාපේක්ෂව පහසුය, අවශ්‍ය වන්නේ බිටුඅනුසාරී ප්‍රතිලෝම පමණි.

Disadvantages of 1’s complement
එකෙහි අනුපූරකයෙහි අවාසි

Two representations exist for zero, which can lead to complications in arithmetic operations and comparisons.
ශුන්‍ය සඳහා නිරූපණයන් දෙකක් පවතී, එය අංක ගණිතමය මෙහෙයුම් සහ සැසඳීම් වලදී සංකූලතා ඇති කළ හැක.

1's complement has a limited range of representable numbers compared to 2's complement
දෙකෙහි අනුපූරකයට සාපේක්ෂව එකෙහි අනුපූරකයට නියෝජනය කළ හැකි සංඛ්‍යා සීමිත පරාසයක් ඇත

2’s complement
2හි අනුපූරකය

Two's complement is an alternative method used to represent positive and negative integers in modern computing systems.
Two's complement යනු නවීන පරිගණක පද්ධතිවල ධන සහ සෘණ නිඛිල නියෝජනය කිරීම සඳහා භාවිතා කරන විකල්ප ක්‍රමයකි.

Two’s complement value for a negative number is found by first finding the One’s complement and adding 1 to it.
සෘණ සංඛ්‍යාවක් සඳහා දෙකෙහි අනුපූරක අගය සොයාගනු ලබන්නේ පළමුව එකෙහි අනුපූරකය සොයාගෙන එයට 1ක් එකතු කිරීමෙනි.

Let’s consider 2-bit two’s complement.
බිටු 2 ක, දෙකෙහි අනුපූරකය සලකා බලමු.

00	is normally 0	but now +0
01	is normally 1	but now +1
10	is normally 2	but now -2
11	is normally 0	but now -1

Let’s consider 3 bit two’s complement.
බිටු 2 ක, තුනෙහි අනුපූරකය සලකා බලමු.

000	is normally 0	but now +0
001	is normally 1	but now +1
010	is normally 2	but now +2
011	is normally 3	but now +3
100	is normally 4	but now -4
101	is normally 5	but now -3
110	is normally 6	but now -2
111	is normally 7	but now -1

As the standard number of bits used to represent one's complement numbers in the syllabus is 8-bits, either positive or negative, the binary number should be written in 8-bits.
විෂය නිර්දේශය තුළ දෙකෙහි අනුපූරක සංඛ්‍යා නියෝජනය කිරීමට භාවිතා කරන සම්මතය බිටු 8 සංඛ්‍යා බැවින්, ධන හෝ සෘණ ඕනෑම ද්විමය අංකයක් බිටු 8 කින් ලිවිය යුතුය.

Using this method, we can represent both negative and positive values using 8 bits ranging from -128 up to +127 instead of representing positive numbers ranging from 0 up to 255.
බිටු 8 ක් භාවිතා කරමින් මෙම ක්‍රමය යෙදූ විට, අපට 0 සිට 255 දක්වා වූ ධන සංඛ්‍යා නියෝජනය කරනවා වෙනුවට -128 සිට +127 දක්වා වූ සෘණ සහ ධන අගයන් දෙකම නිරූපණය කළ හැක.

in 8-bit two’s complement
Positive 5 is 00000101.
Negative 5 is 11111011.

If we take less than or more than 8 bits the range will differ as follows
බිටු 8කට වඩා අඩුවෙන් හෝ වැඩි වශයෙන් ලබාගත් විට පරාසය පහත පරිදි වෙනස් වේ.

1 bit	incompatible
2 bits	-2 to +1
3 bits	-4 to +3
4 bits	-8 to +7
5 bits	-16 to +15
6 bits	-32 to +31
7 bits	-64 to +63
8 bits	-128 to +127
9 bits	-256 to +255
10 bits	-512 to + 511

Example 01
n1-n2 (n1>n2)

	1	0	1	0	0	1	0	1_2	n1
-	0	1	1	1	0	1	1	0_2	n2
	0	0	1	0	1	1	1	1_2	

1	1	1	1	1	1	1	1_2	
0	1	1	1	0	1	1	0_2	n2
1	0	0	0	1	0	0	1_2	n3 [1’s com]
						+	1	
1	0	0	0	1	0	1	0_2	n4 [2’s com]

NOT gate

	1	0	1	0	0	1	0	1_2	n1
+	1	0	0	0	1	0	1	0_2	n4
1	0	0	1	0	1	1	1	1_2	

overflow

Example 02
n1-n2 (n1=n2)

=5 – 5 → 5 + (- 5)

+5 →	0	0	0	0	0	1	0	1_2	n1
-5 →	Invert +5 & add 1								
	1	1	1	1	1	0	1	0_2	
							+	1	
	1	1	1	1	1	0	1	1	n2 [2’s com]

	0	0	0	0	0	1	0	1_2	n1
+	1	1	1	1	1	0	1	1_2	n2
1	0	0	0	0	0	0	0	0_2	

overflow

Example 03

n1-n2 (n1<n2)

= 20 – 35 → ?

= 20 + (-35)

+20 → 0 0 0 1 0 1 0 0_2 — n1

+35 → 0 0 1 0 0 0 1 1_2

→ Invert +35 & add 1

1 1 0 1 1 1 0 0_2

\+ 1

1 1 0 1 1 1 0 1_2 — n2

[2's com]

0 0 0 1 0 1 0 0_2 — n1

\+ 1 1 0 1 1 1 0 1_2 — n2

1 1 1 1 0 0 0 1_2

Advantages of 2's complement

දෙකෙහි අනුපූරකයෙහි වාසි

Only one representation of zero is provided in two's complement (0000 0000 in an 8-bit system), which simplifies arithmetic and comparisons, eliminating the double zero issue present in one's complement.

දෙකේ අනුපූරකයේ (බිටු 8 පද්ධතියක 00000000) ශුන්‍යයේ එක් නිරූපණයක් පමණක් සපයා ඇත, එය අංක ගණිතය සහ සැසඳීම් සරල කරයි, එකෙහි අනුපූරකයේ ඇති ද්විත්ව ශුන්‍ය ගැටලුව ඉවත් කරයි.

Arithmetic in two's complement is straightforward, as subtraction can be performed by simply adding the two's complement of a number.

සංඛ්‍යාවක දෙකේ අනුපූරකය සරලව එකතු කිරීමෙන් අඩු කිරීම සිදු කළ හැකි බැවින් දෙකෙහි අනුපූරකයේ අංක ගණිතය සරල ය.

Disadvantages of 2's complement

දෙකෙහි අනුපූරකයෙහි අවාසි

Calculating the two's complement of a number is complex compared to calculating the one's complement.

සංඛ්‍යාවක දෙකෙහි අනුපූරකය ගණනය කිරීම එකෙහි අනුපූරකයට සාපේක්ෂව සංකීර්ණය.

If the result of an arithmetic operation exceeds the representable range of the number system, overflow can occur, leading to incorrect results.

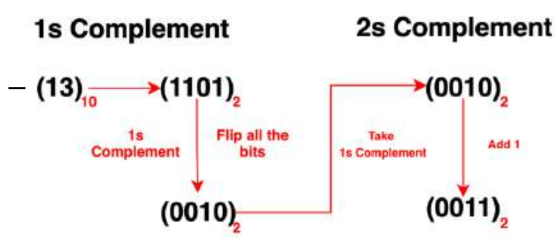
අංක ගණිතමය මෙහෙයුමක ප්‍රතිඵලය සංඛ්‍යා පද්ධතියේ නිරූපණය කළ හැකි පරාසය ඉක්මවා ගියහොත්, overflow සිදු විය හැකි අතර, එය වැරදි ප්‍රතිඵලවලට මග පාදයි.

The range of integers that can be represented is asymmetric.

නිරූපණය කළ හැකි නිඛිල පරාසය අසමමිතික වේ.

Negative number in 2's complement requires inverting all bits and adding 1, which can be more complex than in 1's complement.

දෙකෙහි අනුපූරකයේ දී සෘණ සංඛ්‍යාවක් සඳහා සියලුම බිටු ප්‍රතිලෝම කර 1 ක් එකතු කිරීමට අවශ්‍ය වේ, එය එකෙහි අනුපූරකයට වඩා සංකීර්ණ විය හැක.



Comparison
සංසන්දනය

3-bit numbers

Unsigned magnitude		invert	Signed magnitude		invert	One's complement			two's complement	
000	+0		000	+0		000	+0		000	+0
001	+1		001	+1		001	+1		001	+1
010	+2		010	+2		010	+2		010	+2
011	+3		011	+3		011	+3		011	+3
100	+4		100	- 0		100	- 3		100	- 4
101	+5		101	- 1		101	- 2	100 + 1	101	- 3
110	+6		110	- 2		110	- 1	101 + 1	110	- 2
111	+7		111	- 3		111	0	110 + 1	111	- 1

4-bit numbers

Unsigned magnitude		invert	Signed magnitude		invert	One's complement			two's complement	
0000	+0		0000	+0		0000	+0		0000	+0
0001	+1		0001	+1		0001	+1		0001	+1
0010	+2		0010	+2		0010	+2		0010	+2
0011	+3		0011	+3		0011	+3		0011	+3
0100	+4		0100	+4		0100	+4		0100	+4
0101	+5		0101	+5		0101	+5		0101	+5
0110	+6		0110	+6		0110	+6		0110	+6
0111	+7		0111	+7		0111	+7		0111	+7
1000	+8		1000	- 0		1000	- 7		1000	- 8
1001	+9		1001	- 1		1001	- 6	1000 + 1	1001	- 7
1010	+10		1010	- 2		1010	- 5	1001 + 1	1010	- 6
1011	+11		1011	- 3		1011	- 4	1010 + 1	1011	- 5
1100	+12		1100	- 4		1100	- 3	1011 + 1	1100	- 4
1101	+13		1101	- 5		1101	- 2	1100 + 1	1101	- 3
1110	+14		1110	- 6		1110	- 1	1101 + 1	1110	- 2
1111	+15		1111	- 7		1111	- 0	1110 + 1	1111	- 1

Comparison between methods of negative number representation
සෘණ සංඛ්‍යා නිරූපණය කිරීමේ ක්‍රම අතර සංසන්දනය

Feature **ලක්ෂණ**	**Unsigned Representation**	**Signed Magnitude Representation**	**1's Complement**	**2's Complement**
Representation of positive numbers **ධන සංඛ්‍යා නිරූපනය කිරීම**	Represented directly in binary ද්විමය වශයෙන් සෘජුවම නිරූපනය කරයි	Represented directly in binary with a 0 as the sign bit sign බිටුව 0 ලෙෂ ගෙන ද්විමය වශයෙන් සෘජුවම නිරූපනය කරයි	Represented directly in binary ද්විමය වශයෙන් සෘජුවම නිරූපනය කරයි	Represented directly in binary ද්විමය වශයෙන් සෘජුවම නිරූපනය කරයි
Representation of negative numbers **සෘණ සංඛ්‍යා නිරූපනය කිරීම**	Not applicable අදාළ නොවේ	Represented by inverting the sign bit to 1 sign බිටුව 1 ට පෙරළීමෙන් නිරූපනය කරයි	Represented by inverting all bits of the positive number ධන සංඛ්‍යාවේ සියලුම බිටු ප්‍රතිලෝම කිරීමෙන් නිරූපනය කරයි	Represented by inverting all bits of the positive number and adding 1 ධන සංඛ්‍යාවේ සියලුම බිටු ප්‍රතිලෝම කර 1ක් එකතු කිරීමෙන් නිරූපනය කරයි
Number of representations for zero **බිංදුව සඳහා නිරූපනය ගණන**	1	2 (+0 and -0)	2 (+0 and -0)	1
Range of representable numbers නිරූපණය කළ හැකි සංඛ්‍යා පරාසය	Limited to non-negative numbers සෘණ නොවන සංඛ්‍යා වලට සීමා වේ	Limited range due to sign bit sign බිටුව නිසා සීමිත පරාසයක් පවතී	Limited range due to double zero representation ද්විත්ව ශුන්‍ය නිරූපනය හේතුවෙන් සීමිත පරාසයක්	Wider range compared to 1's complement එකෙහි අනුපූරකයට සාපේක්ෂව පුළුල් පරාසයක් පවතී
Hardware implementation **දෘඩාංග ක්‍රියාත්මක කිරීම**	Simpler වඩා සරලයි	More complex than unsigned but simpler than 1's complement ලකුණු රහිත නිරූපනයට වඩා සංකීර්ණ නමුත් එකෙහි අනුපූරකයට වඩා සරල වේ	More complex than unsigned but simpler than 2's complement ලකුණු රහිත නිරූපනයට වඩා සංකීර්ණ නමුත් දෙකෙහි අනුපූරකයට වඩා සරල වේ	Simplest and most efficient සරලම හා වඩාත්ම කාර්යක්ෂම වේ

Fixed-point Representation vs Floating-point Representation

ස්ථාවර ලක්ෂ්‍ය නිරූපණයන් සහ ඉපිලුම් ලක්ෂ්‍ය නිරූපණයන්

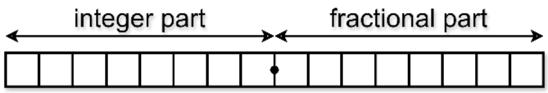
1. Fixed-point Representation

ස්ථාවර ලක්ෂ්‍ය නිරූපණය

In fixed-point representation, numbers are represented with a fixed number of digits before and after the floating point (like the decimal point in the decimal system but in any number base).
ස්ථාවර ලක්ෂ්‍ය නිරූපණයේදී, ඉපිලුම් ලක්ෂ්‍යයට පෙර සහ පසු ස්ථාවර ඉලක්කම් සංඛ්‍යාවක් සමඟ සංඛ්‍යා නිරූපණය කෙරේ (මෙම ක්‍රමය දශමය සංඛ්‍යා පද්ධතියේ දශම ලක්ෂ්‍යය මෙන්ය. නමුත් ඕනෑම සංඛ්‍යා පාදයක පැවතිය හැක).

The precision is constant, meaning the number of fractional bits is fixed.
නිරවද්‍යතාව නියත වේ, එනම් භාගික බිටු ගණන ස්ථාවර වේ.

The range of numbers that can be represented is limited by the number of places reserved for the integer and fractional parts.
නිරූපණය කළ හැකි සංඛ්‍යා පරාසය පූර්ණ සංඛ්‍යා සහ භාගික කොටස් සඳහා වෙන් කර ඇති ස්ථාන ගණනින් සීමා වේ.



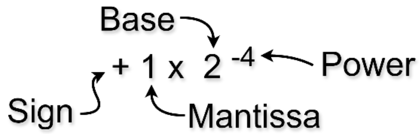
Arithmetic operations using fixed-point numbers are typically faster than those using floating-point numbers.
ස්ථාවර ලක්ෂ්‍ය සංඛ්‍යා භාවිතා කරන අංක ගණිත මෙහෙයුම් සාමාන්‍යයෙන් ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා භාවිතා කරන ඒවාට වඩා වේගවත් වේ.

Fixed-point is often used in embedded systems, digital signal processing, and applications where performance is a concern, and the range of numbers is known in advance.
ස්ථාවර ලක්ෂ්‍යය නිරූපණයන් බොහෝ විට නිහිත පද්ධති, ඩිජිටල් සංඥා සැකසීම සහ කාර්ය සාධනය ගැන සැලකිලිමත් වන යෙදුම්වල භාවිතා වන අතර මේවායේ සංඛ්‍යා පරාසය කල්තියා දැනගත හැකිය.

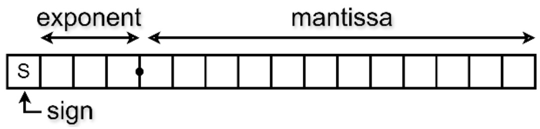
2. Floating-point Representation

ඉපිලුම් ලක්ෂ්‍ය නිරූපණයන්

In floating-point representation, numbers are represented using scientific notation.
ඉපිලුම් ලක්ෂ්‍ය නිරූපණයේදී, විද්‍යාත්මක අංකනය භාවිතයෙන් සංඛ්‍යා නිරූපණය කෙරේ.



sign × mantissa × base$^{\text{exponent}}$
ලකුණ × ප්‍රමාණය × පාදය $^{\text{බලය}}$



Floating-point can represent very large and very small numbers by adjusting the exponent.
ඉපිලුම් ලක්ෂ්‍යය වලට බලය සීරුමාරු කිරීමෙන් ඉතා විශාල සහ ඉතා කුඩා සංඛ්‍යා නියෝජනය කළ හැක.

Precision can vary as numbers get larger or smaller, because the difference between consecutive representable numbers changes.
අඛණ්ඩව නිරූපණය කළ හැකි සංඛ්‍යා අතර වෙනස නිසා සංඛ්‍යා විශාල හෝ කුඩා වන විට නිරවද්‍යතාවය වෙනස් විය හැක.

Floating-point arithmetic is generally slower than fixed-point, especially on processors without a dedicated floating-point unit.
ඉපිලුම් ලක්ෂ්‍ය අංක ගණිතය සාමාන්‍යයෙන් ස්ථාවර ලක්ෂ්‍යයට වඩා මන්දගාමී වේ (විශේෂයෙන් ඒ වෙනුවෙන් වෙන්වූ පාවෙන ලක්ෂ්‍ය ඒකකයක් නොමැති සකසන මත).

Modern CPUs have dedicated sophisticated floating-point units.
නවීන CPU වල විශේෂිත වූ ඉපිලුම් ලක්ෂ්‍ය ඒකක ඇත.

Floating-point operations can introduce rounding errors, which can accumulate in iterative calculations.

ඉපිලුම් ලක්ෂ්‍ය මෙහෙයුම් මඟින් වැටයීමේ දෝෂ හඳුන්වා දිය හැකි අතර ඒවා පුනරාවර්තන ගණනය කිරීම් වලදී එකතු විය හැකිය.

Comparison between Fixed point representation and Floating-point representation
ස්ථාවර ලක්ෂ්‍ය නිරූපණය හා ඉපිලුම් ලක්ෂ්‍ය නිරූපණය අතර සංසන්දනය

Fixed-point representation **ස්ථාවර ලක්ෂ්‍ය නිරූපණය**	**Floating-point representation** **ඉපිලුම් ලක්ෂ්‍ය නිරූපණය**
Has constant precision. නියත නිරවද්‍යතාවයක් පවතී.	Has varying precision. වෙනස් වන නිරවද්‍යතාවයක් පවතී.
Has limited range. සීමිත පරාසයක් ඇත.	Has a wide range. පුළුල් පරාසයක් ඇත.
Less complicated. සංකීර්ණ බව අඩුය.	Complexity is high. සංකීර්ණ බව ඉහළය.
Suitable for applications with known numeric ranges and performance constraints. දන්නා සංඛ්‍යාත්මක පරාසයන් සහ කාර්ය සාධන සීමා සහිත යෙදුම් සඳහා සුදුසු වේ.	Suitable for a wider range of applications but requires more care in implementation. පුළුල් පරාසයක යෙදුම් සඳහා සුදුසු නමුත් ක්‍රියාත්මක කිරීමේදී වැඩි සැලකිල්ලක් අවශ්‍ය වේ.
Can run on typical hardware. සාමාන්‍ය දෘඩාංගවල ක්‍රියාත්මක කළ හැකිය.	Requires specialized hardware to run. ක්‍රියාත්මක කිරීමට විශේෂිත දෘඩාංග අවශ්‍ය වේ.

The IEEE 754 standard
IEEE 754 ප්‍රමිතිය

This is a technical standard established by the Institute of Electrical and Electronics Engineers (IEEE) for representing floating-point numbers in modern computers and for performing arithmetic operations on them.
මෙය නවීන පරිගණකවල ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා නියෝජනය කිරීම සහ ඒවායේ අංක ගණිතමය මෙහෙයුම් සිදු කිරීම සඳහා විද්‍යුත් හා ඉලෙක්ට්‍රොනික ඉංජිනේරු ආයතනය (IEEE) විසින් ස්ථාපිත කරන ලද තාක්ෂණික ප්‍රමිතියකි.

Components of IEEE 754
IEEE 754 හි සංරචක

Sign Bit
සංඛ්‍යාවේ ලකුණ

Specify the sign of the number. Typically, 0 is positive, and 1 is negative.
සංඛ්‍යාවේ ලකුණ සඳහන් කරයි. සාමාන්‍යයෙන්, 0 ධන වන අතර 1 සෘණ වේ.

Exponent
සංඛ්‍යාවේ බලය (ඝාතය)

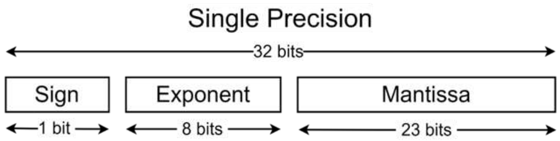
Represents the power to which the base (in binary, the base is 2) is raised.
පාදය (ද්විමය වලදී, පාදය 2 වේ) නංවනු ලබන බලය නියෝජනය කරයි.

Mantissa (or Fraction)
ප්‍රමාණය

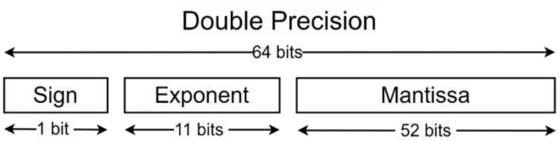
Holds the significant digits of the number.
සංඛ්‍යාවේ ප්‍රමාණාත්මක වටිනාකම නියෝජනය කරන ඉලක්කම් රඳවා තබා ගනී.

IEEE 754 provides several formats:
IEEE 754 ආකෘති කිහිපයක් සපයයි

Single Precision
තනි නිරවද්‍යතාව
(32 bits)



Double Precision
ද්විත්ව නිරවද්‍යතා
(64 bits)



In IEEE notation, the Sign bit and the mantissa is represented directly after converting it into a binary value.
IEEE අංකනයේදී, Sign bit සහ mantissa ද්විමය අගයක් බවට පරිවර්තනය කිරීමෙන් පසු සෘජුවම නිරූපණය කෙරේ.

But the exponent part is represented using the excess K notation. This ensures that the exponent is always stored as a positive number, even though it can represent both positive and negative exponents.
නමුත් ඝාතය (පාදයේ බලය) කොටස excess K අංකනය භාවිතයෙන් නිරූපණය කෙරේ. ධන සහ සෘණ බල දෙකම නියෝජනය කරන සංඛ්‍යා පැවතිය හැකි වුවද, බලය සැමවිටම ධන සංඛ්‍යාවක් ලෙස ගබඩා කර ඇති බව මෙයින් සහතික කෙරේ.

For single-precision (32-bit) floating-point numbers, the bias is 127, while for double-precision (64-bit) numbers, the bias is 1023.
තනි නිරවද්‍යතා (බිටු 32) ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා සඳහා, නැඹුරුව 127 වන අතර ද්විත්ව නිරවද්‍යතා (බිටු 64) සංඛ්‍යා සඳහා, නැඹුරුව 1023 වේ.

Excess-K representation
Biased representation

This is a method used to represent signed binary exponents, primarily for floating-point number representations.
මෙය මූලික වශයෙන් ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා නිරූපණය කිරීමේදී ලකුණු සහිත ද්විමය බලයක් නියෝජනය කිරීමට භාවිතා කරන ක්‍රමයකි.

The "K" value represents the bias added to the actual exponent to make it non-negative.
“K“ අගය මගින් යම් බලයක් සෘණ නොවන බවට පත් කිරීම සඳහා එයට එකතු කරන ලද bias එක පෙන්නුම් කරයි.

The actual value of the number is obtained by subtracting K from the value.
නිරූපිත අගයෙන් K අඩු කිරීමෙන් සංඛ්‍යාවේ සැබෑ අගය ලැබේ.

The main purpose of using Excess-K is to simplify the comparison of floating-point numbers, big or small.
Excess-K නිරූපණයක් භාවිතා කිරීමේ ප්‍රධාන අරමුණ වන්නේ කුඩා හෝ විශාල ඉපිලුම් ලක්ෂ්‍ය සංඛ්‍යා සංසන්දනය කිරීම සරල කිරීමයි.

Using a bias turns all exponent values into positive numbers (unsigned).
නැඹුරුවක් භාවිතා කිරීමෙන් සියලුම ඝාතීය අගයන් ධන සංඛ්‍යා බවට පත් කරයි (ලකුණු රහිත).

This makes it easier to compare floating-point numbers.
මෙමගින් පාවෙන ලක්ෂ්‍ය සංඛ්‍යා සංසන්දනය කිරීම පහසු කරයි.

Example with Excess-127
(Used in IEEE 754 for single-precision floating-point numbers)

Sign bit: 1 bit
Exponent: 8 bits
(With an excess-127 representation)
Significand (or fraction): 23 bits
Total =32 bits

Let's say the actual exponent is +3.
සංඛ්‍යාවේ බලය +3 ලෙස ගනිමු.

1. Using excess-127 representation, add the bias of 127 to the exponent
excess-127 නිරූපණය භාවිතා කරමින්, බලයට 127 එකතු කරන්න.

= 3+127 = 130

2. Convert 130 to binary.
130 ද්විමය බවට පරිවර්තනය කරන්න.

= 10000010

3. This binary value 10000010 is how the exponent would be represented in the IEEE 754 single precision format for a number with an actual exponent of +3.
මෙම 10000010 ද්විමය අගය යනු සැබෑ බලය +3 වන සංඛ්‍යාවක් සඳහා IEEE 754 තනි නිරවද්‍යතා ආකෘතියෙන් බලය කොටස නිරූපණය වන ආකාරයයි.

To decode the exponent
බලය විකේතනය කිරීමට

1. Convert the 8-bit exponent 10000010 from binary to decimal.
10000010 බිටු-8 බලය ද්විමය සිට දශමයට පරිවර්තනය කරන්න.

= 130

2. Subtract 127 from it. This gives the original exponent of +3.
එයින් 127 අඩු කරන්න. මෙය මුල් බලය +3 ලබා දෙයි.

= 130−127 = 3

Let's say the actual exponent is -3.
සංඛ්‍යාවේ බලය -3 ලෙස ගනිමු.

1. Using excess-127 representation, you would add the bias of 127 to the exponent.
excess-127 නිරූපණය භාවිතා කරමින්, බලයට 127 එකතු කරන්න.

= (−3) +127 = 124

2. Convert 124 to binary.
124 ද්විමය බවට පරිවර්තනය කරන්න.

= 01111100.

3. This binary value 01111100 is how the exponent would be represented in the IEEE 754 single precision format for a number with an actual exponent of −3.
මෙම ද්විමය 01111100 අගය යනු සැබෑ බලය -3 වන සංඛ්‍යාවක් සඳහා IEEE 754 තනි නිරවද්‍යතා ආකෘතියෙන් බලය කොටස නිරූපණය වන ආකාරයයි.

To decode the exponent
බලය විකේතනය කිරීමට

1. Convert the 8-bit exponent 01111100 from binary to decimal, which gives 124.
01111100 බිටු-8 බලය ද්විමය සිට දශමයට පරිවර්තනය කරන්න.

= 124

2. Subtract 127 from it. This gives the original exponent of −3.
එයින් 127 අඩු කරන්න. මෙය මුල් බලය −3 ලබා දෙයි

= 124−127 = −3

Example with Excess-1023
(Used in IEEE 754 for double-precision floating-point numbers)

Sign bit: 1 bit
Exponent: 11 bits
(With an excess-1023 representation)
Significand (or fraction): 52 bits
Total =64 bits

Let's say the actual exponent is +3.
සංඛ්‍යාවේ බලය +3 ලෙස ගනිමු.

1. Using excess-1023 representation, you would add the bias of 1023 to the exponent.
excess-1023 නිරූපණය භාවිතා කරමින්, බලයට 1023 එකතු කරන්න.

= 3+1023 = 1026

2. Convert 1026 to binary.
1026 ද්විමය බවට පරිවර්තනය කරන්න.

= 10000000010

3. This binary value 10000000010 is how the exponent would be represented in the IEEE 754 double precision format for a number with an actual exponent of +3.
මෙම 10000000010 ද්විමය අගය යනු සැබෑ බලය +3 වන සංඛ්‍යාවක් සඳහා IEEE 754 ද්විත්ව නිරවද්‍යතා ආකෘතියෙන් බලය කොටස නිරූපණය වන ආකාරයයි.

To decode the exponent
බලය විකේතනය කිරීමට

1. Convert the 11-bit exponent 10000000010 from binary to decimal, which gives 1026.
10000000010 බිටු-11 බලය ද්විමය සිට දශමයට පරිවර්තනය කරන්න.

= 1026

2. Subtract 1023 from it. This gives the original exponent of +3.
එයින් 1023 අඩු කරන්න. මෙය මුල් බලය +3 ලබා දෙයි. = 1026−1023 = 3

Now, the actual exponent is -3.
දැන් සංඛ්‍යාවේ බලය -3 ලෙස ගනිමු.

1. Using excess-1023 representation, you add the bias of 1023 to the exponent, resulting in: −3+1023=1020.
excess-1023 නිරූපණය භාවිතා කරමින්, බලයට 1023 එකතු කරන්න.

= -3+1023 = 1020

2. Convert 1020 to binary.
1020 ද්විමය බවට පරිවර්තනය කරන්න.

= 01111111100

This binary value 01111111100 is how the exponent would be represented in the IEEE 754 double-precision format for a number with an actual exponent of −3.
මෙම ද්විමය 01111111100 අගය යනු සැබෑ බලය −3 වන සංඛ්‍යාවක් සඳහා IEEE 754 ද්විත්ව නිරවද්‍යතා ආකෘතියෙන් බලය කොටස නිරූපණය වන ආකාරයයි.

To decode the exponent
බලය විකේතනය කිරීමට

1. Convert the 11-bit exponent 01111111100 from binary to decimal, which gives 1020.
01111111100 බිටු-11 බලය ද්විමය සිට දශමයට පරිවර්තනය කරන්න.

= 1020

2. Subtract 1023 from it. This gives the original exponent of −3.
එයින් 1023 අඩු කරන්න. මෙය මුල් බලය -3 ලබා දෙයි.

= 1020−1023 = −3

Example 01

Representation of the value +0.15625 in binary form using IEEE 754 single precision format.
IEEE 754 තනි නිරවද්‍යතා ආකෘතිය භාවිතයෙන් +0.15625 අගය ද්විමය ආකාරයෙන් නිරූපණය කිරීම.

1. Convert to binary.
ද්විමය බවට පරිවර්තනය කරන්න.

	.15625	x2
0	.31250	x2
0	.62500	x2
1	.25000	x2
0	.50000	x2
1	.00000	

$0.15625_{10} = 0.00101_2$

2. Write in scientific notation.
විද්‍යාත්මක අංකනයෙන් ලියන්න.

The exponent is applied by the number of places the decimal point moves in forming the required number.
අවශ්‍ය සංඛ්‍යාව සෑදීමේදී දශම තිත චලනය වන ස්ථාන ගණන අනුව ඝාතය යොදනු ලැබේ.

$0\ .\ 0\ \ 0\ \ 1\ \ 0\ \ 1_2$

$0.00101_2 = 1.01 \times 2^{-3}$

3. Determine Sign, Exponent, and Mantissa.
ලකුණ, ඝාතය සහ ප්‍රමාණය තීරණය කරන්න.

Since the number is a positive value, the sign bit is 0.
සංඛ්‍යාව ධන අගයක් බැවින් ලකුණු බිටුව 0 වේ.

Using excess-127 representation, you would add the bias of 127 to the exponent.
excess-127 නිරූපණය භාවිතා කරමින්, බලයට 127 එකතු කරන්න.

Exponent = -3+127 = 124
Now this needs to be converted to a binary number of 8 bits.
දැන් මෙය බිටු 8හි ද්විමය සංඛ්‍යාවක් බවට පත් කරගත යුතුය.

124_{10}

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	1	1	1	1	1	0	0

$124_{10} = 01111100_2$

Exponent = 01111100

The mantissa is found by writing all the bits after the decimal point in the scientific notation until 23 bits are complete.
ප්‍රමාණය සොයාගනු ලබන්නේ විද්‍යාත්මක අංකනයෙහි දශම තිතට පසුව පවතින බිටු සියල්ල බිටු 23 සම්පූර්ණ වන තෙක් ලිවීමෙනි.

Mantissa = 01000000000000000000000

4. Combine All Parts.
සියලුම කොටස් ඒකාබද්ධ කරන්න.

Sign = 0
Exponent = 01111100
Mantissa = 01000000000000000000000

Answer =
0 01111100 01000000000000000000000

Example 02

Representation of the value -8.5 in binary form using IEEE 754 single precision format.
IEEE 754 තනි නිරවද්‍යතා ආකෘතිය භාවිතයෙන් -8.5 අගය ද්විමය ආකාරයෙන් නිරූපණය කිරීම.

1. Convert to binary.
ද්විමය බවට පරිවර්තනය කරන්න.

$8_{10} \rightarrow 1000_2$

	.5	x2
1	.0	

$8.5_{10} = 1000.1_2$

2. Write in scientific notation.
විද්‍යාත්මක අංකනයෙන් ලියන්න.

The exponent is applied by the number of places the decimal point moves in forming the required number.
අවශ්‍ය සංඛ්‍යාව සෑදීමේදී දශම තිත චලනය වන ස්ථාන ගණන අනුව ඝාතය යොදනු ලැබේ.

1 0 0 0 . 1

$1000.1_2 = 1.0001 \times 2^3$

3. Determine Sign, Exponent, and Mantissa.
ලකුණ, ඝාතය සහ ප්‍රමාණය තීරණය කරන්න.

Since the number is a negative value, the sign bit is 1.
සංඛ්‍යාව ඍණ අගයක් බැවින් ලකුණු බිටුව 1 වේ.

Using excess-127 repres පාද entation, you would add the bias of 127 to the exponent.
excess-127 නිරූපණය භාවිතා කරමින්, බලයට 127 එකතු කරන්න.

Exponent = 3+127 = 130

Now this needs to be converted to a binary number of 8 bits.
දැන් මෙය බිටු 8හි ද්විමය සංඛ්‍යාවක් බවට පත් කරගත යුතුය.

130_{10}

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

1	0	0	0	0	0	1	0
$130_{10} = 10000010_2$							

Exponent = 10000010

The mantissa is found by writing all the bits after the decimal point in the scientific notation until 23 bits are complete.
ප්‍රමාණය සොයාගනු ලබන්නේ විද්‍යාත්මක අංකනයෙහි දශම තිතට පසුව පවතින බිටු සියල්ල බිටු 23 සම්පූර්ණ වන තෙක් ලිවීමෙනි.

Mantissa = 00010000000000000000000

4. Combine All Parts.
සියලුම කොටස් ඒකාබද්ධ කරන්න.

Sign = 1
Exponent = 10000010
Mantissa = 00010000000000000000000

Answer =
1 10000010 00010000000000000000000

Example 03

Representation of the value +10.25 in binary form using IEEE 754 double precision format.
IEEE 754 ද්විත්ව නිරවද්‍යතා ආකෘතිය භාවිතයෙන් +10.25 අගය ද්විමය ආකාරයෙන් නිරූපණය කිරීම.

1. Convert to binary.
ද්විමය බවට පරිවර්තනය කරන්න.

$10_{10} \rightarrow 1010_2$

	.25	x2
0	.50	x2
1	.00	

$10.25_{10} = 1010.01_2$

2. Write in scientific notation.
විද්‍යාත්මක අංකනයෙන් ලියන්න.

The exponent is applied by the number of places the decimal point moves in forming the required number.
අවශ්‍ය සංඛ්‍යාව සෑදීමේදී දශම තිත චලනය වන ස්ථාන ගණන අනුව ඝාතය යොදනු ලැබේ.

1 0 1 0 . 0 1_2

$1010.01_2 = 1.01001 \times 2^3$

3. Determine Sign, Exponent, and Mantissa.
ලකුණ, ඝාතය සහ ප්‍රමාණය තීරණය කරන්න.

Since the number is a positive value, the sign bit is 0.
සංඛ්‍යාව ධන අගයක් බැවින් ලකුණු බිටුව 0 වේ.

Using excess-1023 representation, you would add the bias of 1023 to the exponent.
Excess-1023 නිරූපණය භාවිතා කරමින්, බලයට 1023 එකතු කරන්න.

Exponent = 3+1023 = 1026
Now this needs to be converted to a binary number of 11 bits.
දැන් මෙය බිටු 11හි ද්විමය සංඛ්‍යාවක් බවට පත් කරගත යුතුය.

1026_{10}

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1024	512	256	128	64	32	16	8	4	2	1

1	0	0	0	0	0	0	0	0	1	0

10000000010_2

Exponent = 10000000010

The mantissa is found by writing all the bits after the decimal point in the scientific notation until 52 bits are complete.
ප්‍රමාණය සොයාගනු ලබන්නේ විද්‍යාත්මක අංකනයෙහි දශම තිතට පසුව පවතින බිටු සියල්ල බිටු 52 සම්පූර්ණ වන තෙක් ලිවීමෙනි.

Mantissa=
01001000000000000000000000000000000000
00000000000000

4. Combine All Parts.
සියලුම කොටස් ඒකාබද්ධ කරන්න.

Sign = 0
Exponent = 10000000010
Mantissa=010010000000000000000000000000
0000000000000000000000

Answer =
01000000001001001000000000000000000000
00000000000000000000000000

Bitwise operators
බිටුඅනුසාරී මෙහෙයවන

Bitwise NOT

This operation inverts each bit of a binary number. If a bit is 0, it becomes 1, and if it's 1, it becomes 0
මෙම මෙහෙයුම ද්විමය සංඛ්‍යාවක සෑම බිටුවක්ම ප්‍රතිලෝම කරයි. බිටුව 0 නම්, එය 1 බවටත්, 1 නම්, එය 0 බවටත් පත් වේ

Number: 5
Binary: 0101
NOT: 1010

(which is -6 in two's complement representation for signed integers)
(එය ලකුණු සහිත පූර්ණ සංඛ්‍යා සඳහා දෙකේ අනුපූරක නියෝජනයේ -6 ට සමාන වේ)

Example 01
= NOT (178_{10})

N1 = 178_{10}	1	0	1	1	0	0	1	0
NOT(N1)	0	1	0	0	1	1	0	1
NOT (178_{10}) = 01001101_2								

(Which is 77 in decimal)

Example 02
= NOT (74_{10})

N2 = 74_{10}	0	1	0	0	1	0	1	0
NOT(N2)	1	0	1	1	0	1	0	1
NOT (74_{10}) = 10110101_2								

(Which is 181 in decimal)

Bitwise AND

Compares each pair of corresponding bits of two numbers and returns 1 only if both bits are 1.
සංඛ්‍යා දෙකක සෑම අනුරූප බිටු දෙකක්ම සංසන්දනය කර බිටු දෙකම 1 නම් පමණක් 1 ලබා දෙයි; එසේ නොමැති නම්, 0 ලබා දෙයි.

Number A: 5 (Binary: 0101)
Number B: 3 (Binary: 0011)
AND: 0001

(Which is 1 in decimal)

Example 01
= 178_{10} AND 46_{10}

N1 = 178_{10}	1	0	1	1	0	0	1	0
N2 = 46_{10}	0	0	1	0	1	1	1	0
N1 AND N2	0	0	1	0	0	0	1	0
178_{10} AND 46_{10} = 00100010_2								

(Which is 34 in decimal)

Example 02
= 234_{10} AND 74_{10}

N3 = 234_{10}	1	1	1	0	1	0	1	0
N4 = 74_{10}	0	1	0	0	1	0	1	0
N3 AND N4	0	1	0	0	1	0	1	0
234_{10} AND 74_{10} = 01001010_2								

(Which is 74 in decimal)

Bitwise OR

Compares each corresponding bit of two numbers and returns 1 if at least one of the bits is 1.
සංඛ්‍යා දෙකක අනුරූප බිටු දෙක සංසන්දනය කර අවම වශයෙන් බිටු එකක්වත් 1 නම් 1 ආපසු ලබා දෙයි.

Number A: 5 (Binary: 0101)
Number B: 3 (Binary: 0011)
OR: 0111

(Which is 7 in decimal)

Example 01

= 178_{10} OR 46_{10}

N1 = 178_{10}	1	0	1	1	0	0	1	0
N2 = 46_{10}	0	0	1	0	1	1	1	0
N1 OR N2	1	0	1	1	1	1	1	0
178_{10} OR 46_{10} = 10111110_2								

(Which is 190 in decimal)

Example 02

= 234_{10} OR 74_{10}

N3 = 234_{10}	1	1	1	0	1	0	1	0
N4 = 74_{10}	0	1	0	0	1	0	1	0
N3 OR N4	1	1	1	0	1	0	1	0
234_{10} OR 74_{10} = 11101010_2								

(Which is 234 in decimal)

Bitwise XOR

Compares each corresponding bit of two numbers and returns 1 only if the bits are different.
සංඛ්‍යා දෙකක අනුරූප බිටු දෙක සංසන්දනය කර බිටු වෙනස් නම් පමණක් 1 ආපසු ලබා දෙයි.

Number A: 5 (Binary: 0101)
Number B: 3 (Binary: 0011)
XOR: 0110

(Which is 6 in decimal)

Example 01

= 178_{10} XOR 46_{10}

N1 = 178_{10}	1	0	1	1	0	0	1	0
N2 = 46_{10}	0	0	1	0	1	1	1	0
N1 XOR N2	1	0	0	1	1	1	0	0
178_{10} XOR 46_{10} = 10011100_2								

(Which is 156 in decimal)

Example 02

= 234_{10} XOR 74_{10}

N3 = 234_{10}	1	1	1	0	1	0	1	0
N4 = 74_{10}	0	1	0	0	1	0	1	0
N3 XOR N4	1	0	1	0	0	0	0	0
234_{10} XOR 74_{10} = 10100000_2								

(Which is 160 in decimal)