తొమ్మిదో అధ్యాయం

కంప్యూటర్లో డేటాని గుర్తించే పద్ధతి

ఇప్పటి దాకా కంప్యూటర్ లోని ఇన్‌పుట్, ఔట్‌పుట్ యూనిట్‌లెందుకు ఉన్నాయో, సిపియు అంటే ఏమిటో, అవి ఎలా పని చేస్తాయో తెలుసుకున్నాం. కంప్యూటర్ కి మనం ఇచ్చేది ఇన్‌పుట్ అనీ అది మనకు తిరిగి ఇచ్చేది ఔట్‌పుట్ అని కూడా తెలుసుకున్నాం. మనం అందించే సూచనలు అన్నీ ఒక ప్రోగ్రాం రూపంలో కంప్యూటర్ కి అందజేయాలి. ఇదంతా కంప్యూటర్ కి అర్థం అయ్యే 0,1 భాషలోనే అందజేయాల్సి ఉంటుంది. అది ఏబిసిడిలు అయినా సరే. 1,2,3 లు ఐనా సరే. ప్రోగ్రాములని వివిధ రకాల ప్రోగ్రామింగ్ భాషలలో వ్రాయవచ్చు. ఆ సంగతి వేరే అధ్యాయంలో చూద్దాం. ముందు కంప్యూటర్ వాడే రెండు ముఖ్యమైన డేటా టైపులగూర్చి ఈ అధ్యాయంలో తెలుసుకుందాం.

డేటా టైపులు

కంప్యూటర్ కి మనం అందించే డేటా ప్రధానంగా రెండు రకాలు. అవి, 1. అంకెలతో కూడింది. 2. అంకెలు, అక్షరాలతో కూడింది. అంకెలతో కూడిన డేటానే న్యూమరిక్ డేటా(Numeric Data) అంటారు. అంకెలు, అక్షరాలతో కూడిన డేటానే ఆల్ఫా న్యూమరిక్ డేటా(AlphaNumeric Data) అంటారు.

న్యూమరిక్ డేటా

న్యూమరిక్ డేటా అంటే, 0,1,2,3,4,5,6,7,8,9 అన్న పది అంకెల సముదాయంతో ఏర్పడే డేటా. ఈ న్యూమరిక్ డేటా తిరిగి రెండు రకాలు. 1. ఇంటిజర్స్(Integers) లేదా పూర్ణ సంఖ్యలు, 2. రియల్ నంబర్స్(Real Numbers) లేదా నిజ సంఖ్యలు. ఉదాహరణకి, ఉద్యోగుల వయస్సు, మూల వేతనం అనే వివరాలు ఉన్నాయనుకోండి. వయస్సు అనేది పూర్ణ సంఖ్య లేదా ఇంటిజర్. అదే మూల వేతనం అనేది రియల్ నంబర్. అందరు ఉద్యోగుల డేటాని ఒక చోట ఉంచి కంప్యూటర్ కి అందజేయడం అంటే న్యూమరిక్ డేటాని

అందజేయడమే అవుతుంది. ఈ న్యూమరిక్ డేటాని యథాప్రకారం కూడికలు, తీసివేతలు, గుణకార, భాగహారాలు - అన్నిటిలో వాడటం జరుగుతుంది.

ఆల్ఫాన్యూమరిక్ డేటా

ఆల్ఫా న్యూమరిక్ డేటా అంటే, 0,1,2,3,4,5,6,7,8,9 అన్న పది అంకెల సముదాయం మాత్రమే కాదు. A నించి Z, a నించి z దాకా అన్ని అక్షరాలు, !,@,#,$,%,^,&,*,(,),_,+,|,? అన్ని రకాల ప్రత్యేక గుర్తులు అన్నీ కలిసి ఏర్పడే డేటా. పైన చెప్పుకున్న ఉదాహరణలో ఉద్యోగుల వివరాలు అన్నపుడు ఎంప్లాయీ కోడ్, వయస్సు, మూల వేతనం అని ఉందనుకోండి. ఎంప్లాయీ కోడ్ సాధారణంగా ఆల్ఫాన్యూమరిక్ డేటాగా ఉంటుంది(ఉదా. A00213).

ఈ ఆల్ఫాన్యూమరిక్ డేటానే ఒక్కోసారి కారెక్టర్(Character) డేటా అనీ అంటారు. ఐతే ఈ కారెక్టర్ డేటాలో అంకెలు కూడా అక్షరాల్లా గుర్తింపబడతాయేగానీ అంకెల స్థానాలూ అవీ గుర్తింపబడవు. ఈ కారెక్టర్ డేటాలో ఉండే అంకెలపై ఎలాటి కూడికలు, తీసివేతలు, గుణకార, భాగహారాలూ చేయలేము. ఒక వేళ అలా చెయ్యాల్సి వస్తే, ఆయా డేటాని న్యూమరిక్ గా ప్రోగ్రాం ద్వారా మార్చుకుని తంటాలు పడాల్సి ఉంటుంది.

డేటా, ఇన్ఫర్మేషన్

అంకెలనీ, సంఖ్యలనీ, అక్షరాలనీ అర్థవంతంగా పోగేసి ఒకచోట ఉంచి కంప్యూటర్ కి అందజేయడమే డేటాని కంప్యూటర్ కి ఇన్పుట్గా అందజేయడం అనబడుతుంది. అంటే, కంప్యూటర్ కి మనం ఇచ్చే సూచనలని ప్రోగ్రాం(Program) అనీ, సంబంధిత విలువలని డేటా(Data) అనీ అంటారు. ఈ డేటానే అచ్చ తెనుగులో దత్తము అని అంటారు.

ఒక విషయానికి సంబంధించిన అంకెలు, సంఖ్యలు, అక్షరాలు - వీటి సముదాయమే డేటా అవుతుంది. అంటే, ఉదాహరణకి, ప్రియాంక, 50, 3000, 400 అనేది డేటా. ఇందులో ప్రియాంక అనేది ఒక మనిషి(స్త్రీ) పేరు, 50 ఆమె వయస్సు, 3000 ఆమె మూల వేతనం, 400 అనేది అలవెన్సు. ఇలా ఉద్యోగుల డేటా అంతా ముందే కంప్యూటర్ లో ఫీడ్ చెయబడి ఉందనుకోండి. అపుడు, కంప్యూటర్ కి 3000 రు. మూలవేతనం ఉండే ఆడవారి పేర్లు సూచించమని మన ప్రోగ్రాంలో మనం కోరితే, కంప్యూటర్ దాని

ఫలితాన్ని ఇలా చూపుతుంది.

రు 3000 మూల వేతనం గల ఉద్యోగుల జాబితా

రోజా

ప్రియాంక

స్మిత

శ్రీదేవి

అంటే, 3000రూపాయలు మూల వేతనం ఉండే ఆడవారి పేర్లన్నిటినీ ఇలా అర్థవంతమైన సమాచారంగా ఇవ్వడం జరుగుతుంది. అంచేత దీనిని సమాచారం లేదా ఇన్ఫర్మేషన్(Information) అంటారు. డేటాని శుద్ధిచేయగా వచ్చేది సమాచారం. ఈ శుద్ధిచేయడాన్నే ఆంగ్లంలో ప్రాసెసింగ్(Processing) అంటారు. డేటాని శుద్ధిచేయడం కాబట్టి ఆంగ్లంలో డేటా ప్రాసెసింగ్(Data Processing) అంటారు. కంప్యూటర్ ఎలక్ట్రానిక్స్‌తో ముడిపడి ఉంది గనక ఈ ప్రాసెసింగ్‌ని ఎలక్ట్రానిక్ డేటా ప్రాసెసింగ్ లేదా ముద్దుగా ఇడిపి అంటారు. ఈ ప్రాసెసింగ్‌కి అర్థం అయ్యేలా చెప్పాలంటే, నిత్య జీవితంలో రోజూ మన ఇంట్లో చేసే సాంబారుని ఉదాహరణగా చెప్పుకోవచ్చు. సాంబారు తయారీకి కావలిసిన ముడి పదార్థాలైన పప్పు, ఉప్పు, చింతపండు పులుసు, కూర ముక్కలు వగైరా డేటా ఐతే, దానిని స్టౌ పై ఉంచి సాంబారుగా కాచడం అనేది ప్రాసెసింగ్. ఫలితంగా తయారయ్యేది సాంబారు - ఇన్ఫర్మేషన్ అన్నమాట.

పప్పు, ఉప్పు, చింతపండు పులుసు, కూర ముక్కలు వగైరా → స్టౌ పై ఉంచి, కాచడం లాటి పనులు → సాంబారు

(డేటా) **(ప్రాసెసింగ్)** **(ఇన్ఫర్మేషన్)**

డేటా నించి ఇన్‌ఫర్మేషన్ పొందడం

డేటా అనేదాన్ని ఇన్‌పుట్ అని, ఇన్ఫర్మేషన్‌ని ప్రాసెసింగ్ యొక్క ఔట్‌పుట్ అనీ అంటారు. అంటే, పప్పూ ఇతర సామగ్రి ఇన్‌పుట్ ఐతే, సాంబారు ఔట్‌పుట్. అలాగే, బియ్యాన్ని నీరులో (ఇన్‌పుట్) కలిపి ఉడకబెడితే(ప్రాసెస్), అన్నం (ఔట్‌పుట్) అవుతుంది.

డేటాని కంప్యూటర్ అర్థం చేసుకునే తీరు

కంప్యూటర్ రన్ చేయబోయే ప్రోగ్రాంని, దాని తాలూకు డేటానీ ప్రైమరీ మెమరీలో ఉంచుకుంటుంది. ఈ మెమరీ అనేది బుల్లి బుల్లి అరలు లేదా సెల్స్(cells)గా విడగొట్టబడి ఉంటుంది. ప్రతీ మెమరీ సెల్ ఒక రకమైన స్విచ్చ్ లాగా పనిచేస్తుంది. అంటే, అది 0,1 అనే రెండు అంకెలని మాత్రమే ఉంచుకోగలదు. విద్యుత్ ప్రసరించినపుడు అది ఆన్ అవుతుంది. అంటే 1 అని అర్థం. లేదంటే 0 అన్నమాట. అంటే, మనం అందించే ప్రతీ డేటా ఈ 0,1ల సముదాయాలుగా అందించాల్సిందే. ఇవి రెండు అంకెలు 0,1. రెండుని ఆంగ్లంలో బైనరీ(Binary) అంటారు. రెండు అంకెలు కాబట్టి 0,1లని బైనరీ డిజిట్‌లు(Binary Digits) అంటారు. ఈ Binary Digit లో ఉండే తొలి అక్షరం B చివరి 2 అక్షరాలు itలు వెరసి BIT అయ్యింది. అంటే BIT అనేది Binary Digitకి క్లుప్త రూపం. మనం B అని టైప్ చేసినా అది తిరిగి 0,1 లోకి తర్జుమా కాబడే కంప్యూటర్ అర్థం చేసుకుంటుంది. 0 లేదా 1 అని దాయడాన్ని బిట్ అనీ,. 8 బిట్‌లు 1 బైట్(Byte) అనీ అంటారు.

ఆల్ఫాన్యూమరిక్ లేదా కారెక్టర్ డేటా

ఆంగ్లంలో అక్షరాలు A - Z మొత్తం 26 ఉన్నాయి. మళ్లీ ఇవి కాపిటల్స్(Capital Letters), స్మాల్ (Small Letters) అని రెండు రకాలున్నాయి. ఈ విషయం అందరికీ తెలుసు. వీటినీ, అంకెలనీ, ఇతర ప్రత్యేక గుర్తులనీ మనం 0,1లలో గుర్తించాలీ అంటే దాదాపు 94 అక్షరాలుగా చూపాలి. అలా 0,1లతోనే చెయ్యాలంటే 7 బిట్లతో సూచించవచ్చు. దానికి మనం దాదాపు 2^7 అక్షరాలు వాడాలి. అంటే 128 అన్నమాట. ఐతే, తెలుగు భాషలోని అక్షరాలవంటివి 128 కన్నా ఎక్కువే ఉంటాయి. కాబట్టి 8బిట్‌లతో సూచించవలసి వస్తుంది. అంటే 8బిట్స్‌తో సూచిస్తే, దాదాపు 256 అక్షరాలని($2^8 = 256$) సూచించగలం. ఇలా0,1లతో ఏర్పడిన కోడ్‌లనే బైనరీ కోడ్ అంటారు. ఈ కోడ్‌లని ప్రమాణీకరించింది అమెరికాకి చెందిన సంస్థ. వాటినే ఆస్కీ(ASCII) కోడ్‌లంటారు.

ASCII -అంటే, American Standard Code For Information Interchange అని ఆంగ్ల రూపాంతరం. ASCII-7 బిట్ కోడ్‌లని ఉపయోగించి 128 అక్షరాలని కోడ్ రూపంలో సూచించవచ్చు. అదే ASCII-8 బిట్ కోడ్‌లని ఉపయోగించి 256 అక్షరాలని కోడ్ రూపంలో సూచించవచ్చు. పక్క పేజీలో 7 బిట్, 8 బిట్ కోడ్‌లని A నించీ Z వరకు వివరంగా పొందుపరచబడింది.

CHARACTER	Numeric Value	7-Bit ASCII Code	8-Bit ASCII Code
A	65	100 0001	1010 0001
B	66	100 0010	1010 0010
C	67	100 0011	1010 0011
D	68	100 0100	1010 0100
E	69	100 0101	1010 0101
F	70	100 0110	1010 0110
G	71	100 0111	1010 0111
H	72	100 1000	1010 1000
I	73	100 1001	1010 1001
J	74	100 1010	1010 1010
K	75	100 1011	1010 1011
L	76	100 1100	1010 1100
M	77	100 0011	1010 0011
N	78	100 1101	1010 1101
O	79	100 1111	1010 1111
P	80	101 0000	1010 1111
Q	81	101 0001	1011 0001
R	82	101 0010	1011 0010
S	83	101 0011	1011 0011
T	84	101 0100	1011 0100
U	85	101 0101	1011 0101
V	86	101 0110	1011 0110
W	87	101 0111	1011 0111
X	88	101 1000	1011 1000
Y	89	101 1001	1011 1001
Z	90	101 1010	1011 1010

ఇవన్నీ ఆస్కి ప్రమాణాలలో ఇచ్చిన కోడ్ లు. ఈ 8 బిట్ కోడ్ లే ప్రపంచ ప్రమాణాలుగా ఉన్నాయి. నేడు ఆస్కి కోడ్ లకి మల్లే భారతీయ భాషలకోసం ఇస్కి(ISCII) అనే ప్రమాణాలు అమలులోకి వచ్చాయి. ఇస్కి అంటే Indian Standard Code For Information Interchange అని అర్థం. ఇవి కూడా 8 బిట్ కోడ్ లే. వివిధ అక్షరాలకు ఉండే ఇస్కి విలువల పట్టికని పుస్తకం చివర్న ఇవ్వబడింది.

ఒక్కోసారి అక్షరాలతో బాటు అంకెలూ ఉంటాయి. వీటన్నిటినీ విడిగా అక్షరాలూ, అంకెలూ అని గుర్తించాలీ అంటే అవి కారెక్టర్, డిజిట్స్ విడివిడిగా చదువుకుంటే సరి. కలివిడిగా ఉంటే మాత్రం అవన్నీ అక్షర సముదాయంగానే దాయబడి, గుర్తింపబడతాయి. మరో సంగతి. రెండు పదాల మధ్య ఉండే ఖాళీ కూడా మెమరీలో ఒక బైట్ స్థానాన్ని ఆక్రమిస్తుంది.

న్యూమరిక్ డేటా

ఆంగ్లంలో కడు పొందికగా కేవలం 26 అక్షరాలు ఉన్నాయి. కాపిటల్, స్మాల్ అక్షరాలు కలిసి 52. వాటికి తోడు దాదాపు 30 పైగా ప్రత్యేక గుర్తులున్నాయి. ఇవి కాక ఉన్నవి 0 నించి 9 దాకా 10 అంకెలు. లెక్కలు వేయడంలో ఈ అంకెలకు ఉండే ప్రాముఖ్యాన్ని వేరేగా చెప్పనక్కరలేదు. వీటిని కంప్యూటర్ వేరే రకంగా గుర్తిస్తుంది. అదెలాగో చూద్దాం.

కంప్యూటర్ కేవలం 0 లేదా 1లని మాత్రమే అర్థం చేసుకుంటుందని (అంటే, విద్యుత్ ఆ సర్క్యూట్‌లో ప్రసరిస్తే, 1 అనీ లేకుంటే 0 అనీ) మనకి తెలుసు. ఉదాహరణకి, 9392 అన్న సంఖ్యని తీసుకోండి. ఒకట్ల స్థానంలో 2, పదుల స్థానంలో 9, వందల స్థానంలో 3, వేల స్థానంలో 9. అంటే, 9392 అన్నదానిని $9X10^3+3X10^2+9X10^1+2X10^0$ అని సూచించవచ్చు. ఇలా వ్రాయడాన్ని ఆంగ్లంలో, డెసిమల్ పద్ధతి అనీ, తెనుగులో దశాంశ పద్ధతి అనీ అంటారు. జాగ్రత్తగా ఆలోచిస్తే, ఆ పేరెందుకు పెట్టారన్నది మనకే తెలుస్తుంది. పైన 9392ని 10ని ఆధారంగా (బేస్ లేదా రాడిక్స్ అంటారు) చెబుతున్నాం. 8ని ఆధారంగా 9392 ని సూచిస్తే, $9X8^3+3X8^2+9X\ 8^1+2X\ 8^0$ అని వ్రాయాలి. ఇలా 8 ఆధారంగా సూచించే దాన్ని ఆక్టల్ పద్ధతి లేదా అష్టాంశ పద్ధతి అంటారు.

కేవలం 0,1 ల ప్రసక్తి వచ్చింది కాబట్టి, ఆ పద్ధతి గూర్చి క్లుప్తంగా తెలుసుకుందాం. 10101 అనేది ఓ 2 ని బేస్‌గా వాడి సూచించబడింది. దీని అర్థం ఏంటంటే, $1X\ 2^4+0X2^3+1X2^2\ +0X2^1+1X2^0$. దశాంశ పద్ధతిలో దీనివిలువ 21 ఔతుంది. 2 ని బేస్‌గా వాడాం గనక దీనిని బైనరీపద్ధతి అంటారు. ఈ సంఖ్యలని వేరే

అంకెని ఆధారంగా చేసుకుని కూడా చెప్పచ్చు. (పట్టిక చూడండి).

[illegible]

ఆధారం అంకె(బేస్) 10	8	3	2	12	16
1	1	1	001	1	1
2	2	2	010	2	2
3	3	10	011	3	3
4	4	11	100	4	4
5	5	12	101	5	5
6	6	20	102	6	6
7	7	21	110	7	7
8	10	22	111	8	8
9	11	100	.	9	9
10	12	101	.	X	A
11	13	102	.	Y	B
12	14	110	.	10	C
13	15	111	.	.	D
14	.	.	.	.	E
15	.	.	.	.	F
16	.	.	.	.	10
17	.	.	.	.	11

పైన చెప్పుకున్న పద్ధతులలో ఎక్కువగా కంప్యూటర్ వాడేది బైనరీ, ఆక్టల్ పద్ధతులనూ, 16ని ఆధారంగా పనిచేసే షోడశాంశ(హెక్సాడెసిమల్) పద్ధతి. ఇపుడు ఈ పద్ధతుల మధ్య అంకెల మార్పిడీ ఎలా చేయవచ్చో చూద్దాం. ఉదాహరణకి, 65743 అన్న ఓ అష్టాంశపద్ధతి సంఖ్య ఉందనుకుందాం. దాన్ని దశాంశ పద్ధతిలో మార్చాలీ అంటే, $6X8^4 + 5X8^3 + 7X8^2 + 4X8^1 + 3X8^0$ అని సూచిస్తూ గుణకారాలూ, కూడికలూ చేస్తే, దశాంశ విలువ వస్తుంది.

ఐతే, అన్ని గుణకారాలు చేసేప్పటికి మనకి ఓపిక నశిస్తుంది. అంత శ్రమ పడాలి. ఐతే, దీనిని సులభంగా చేయడానికి పద్ధతులున్నాయి. 65743లో ఉండే 6ని 8తో ముందు గుణించాలి. వచ్చిన 48కి 6 తర్వాత ఉన్న 5 ని కూడండి. అంటే 53 అయింది. ఈ 53ని తిరిగి 8తో గుణించాలి. వచ్చిన 424కి పక్కనే ఉన్న 7 కూడండి. అంటే 431 వస్తుంది. ఈ 431ని 8తో గుణించాలి. దీనికి, పక్కనే ఉన్న 4కూడాలి. అపుడు 3452 వస్తుంది. ఈ 3452ని 8 తో గుణించి, చివరి 3 కూడితే 27619 వస్తుంది. ఇదీ 65743 అనే అష్టాంశ(పద్ధతి) సంఖ్యకి దశాంశ పద్ధతిలో విలువ. ఇట్లే బైనరీ పద్ధతిలో గానీ, మరో దాన్లోగానీ పేర్కొన్న సంఖ్యని దశాంశపద్ధతిలోకి మార్చవచ్చు. మనం ఏ పద్ధతిలో ఐతే సంఖ్యని సూచిస్తున్నామో, ఆ పద్ధతి ఆధారపు

అంకె(బేస్)తో గుణించాలి. అంటే బైనరీనించీ దశాంశపద్ధతిలోకి అనుకోండి. పై ఉదాహరణలో ఆక్టల్ సంఖ్య కాబట్టి, దాన్ని దశాంశపద్ధతిలోకి మార్చేందుకని ఆ సంఖ్యని 8తో గుణించాం.

దశాంశపద్ధతి నించీ బైనరీలోకి పరివర్తనం

	98	శేషం	
98/2	49	0	↑
49/2	24	1	
24/2	12	0	
12/2	6	0	క్రింద నించీ
6/2	3	0	పైకి
3/2	1	1	సూచించాలి.
1/2	0	1	

అదే ఆ సంఖ్య షోడశాంశ పద్ధతిదైతే, 8 స్థానంలో 16 ఉండాలి. అలాగే, దశాంశ పద్ధతిలోనించి బైనరీలోకి లేదా అష్టాంశపద్ధతికి లేదా మరోదానికి మార్చాలన్నా ఇదే తంతు. ఎటొచ్చీ, పైన గుణకారం చేశాం. ఇక్కడ భాగహారం చేయాలి. అంతే.

కేవలం దశాంశ పద్ధతి నించీ బైనరీ పద్ధతిలోకెలా మార్చాలో మాత్రం చూద్దాం దశాంశ పద్ధతిలో ఉండే 98 అన్న సంఖ్యని బైనరీలోకి మార్చాలనుకుందాం. అపుడు మనం చెయ్యాల్సిందెల్లా, 98ని 2తో భాగిస్తూ పోవడమే. ప్రతీసారీ 2తో భాగించి, శేషం విలువని పక్కగా వ్రాసుకోవాలి. అంతే. అంటే 98 అన్న దశాంశ పద్ధతిలోని సంఖ్య విలువ బైనరీ పద్ధతిలో 1100010 అన్నమాట.

అంకెల గారడీ అసలు విషయం!

కంప్యూటర్ది పూర్తిగా 0,1 అనే అంకెల గారడీయే వేదాంతం. ముందు పేజీల్లో, ఎలాటి అంకెల పద్ధతులు ఉన్నాయో, ఒక పద్ధతి నించి మరో పద్ధతిలోకి ఎలా మార్చడమన్నదీ, బైనరీ, ఆక్టల్, డెసిమల్ పద్ధతులంటే ఏమిటో వాటి గూర్చి తెలుసుకున్నాం. వీటికి తోడు 16ని ఆధారసంఖ్యగా అంకెలని సూచించవచ్చని కూడా తెలుసుకున్నాం. ఈ 16ని ఆధారంగా పనిచేసే అంకెల పద్ధతికే షోడశాంశ పద్ధతి లేదా హెక్సాడెసిమల్ పద్ధతి అని పేరు. అంటే, దశాంశ పద్ధతిలో మనం 16 అంటే, షోడశాంశ పద్ధతిలో 10 అన్నమాట. ఒక ఆక్టల్ లేదా ఒక హెక్సా డెసిమల్ పద్ధతిలోని సంఖ్యలని మూడు స్థానాల బైనరీ అంకెల ద్వారా సూచించవచ్చు. దీన్నే బైనరీ ట్రిప్లెట్ అంటారు. బైనరీ ట్రిప్లెట్ అంటే, మూడు స్థానాల బైనరీ అంకె అని అర్థం.

ఆక్టల్ సంఖ్య	బైనరీ ట్రిప్లెట్
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

అంటే, 010101111100101 అనే బైనరీ సంఖ్య ఉందనుకుంటే, దాన్ని పైన చెప్పినట్టు బైనరీ ట్రిప్లెట్‌లుగా వ్రాస్తే , 010 , 101, 111, 100, 101 అని 5 ట్రిప్లెట్‌లు వస్తాయి. వీటిని ఆక్టల్ సంఖ్యలుగా మార్చి వ్రాస్తే, 2, 5, 7, 4, 5 వెరసి 25745 అవుతుంది. దీనిని కంప్యూటరెలా వాడుతుందో చూద్దాం. కంప్యూటర్‌కి 0,1 తప్ప వేరే అంకెలు లేవు. ముఖ్య కారణం కరెంటు సర్క్యూట్లలో ప్రసరించే తీరు. కరెంటు వెళితే 1 లేదూ 0 అని కదా కంప్యూటర్ అర్థం చేసుకునేది. ఇదెంత సులువో, ఎలాటి సంఖ్యనైనా బైనరీలో సూచించడం అంతే సులువు. రెండో కారణం ఏమిటంటే, కంప్యూటర్ల లో ఉండే గణిత శాస్త్రి సర్క్యూట్లు కూడా ఇక 0,1 ల ఆధారంగానే కదా పనిచేసేది. ఈ రెండు కారణాలూ ఒకదానికొకటి పెనవేసుకుని ఉన్నాయి.

అందువల్ల దశాంశ సంఖ్యలూ, ఆక్టల్ సంఖ్యలూ వాడేకన్నా వాటిని ఒకేసారి బైనరీలోకి మార్చి దానిమీద పనిచేయడం సులభం. అందుకే కంప్యూటర్లలో, ఆక్టల్‌కీ హెక్సాడెసిమల్‌కీ ఎంతో ప్రాముఖ్యత ఉంది. కంప్యూటర్ వేగం కూడా దానిపైనే ఆధారపడి ఉంది. ఒక్క ఆక్టల్ సంఖ్యనే కాదు హెక్సాడెసిమల్ సంఖ్యని కూడా బైనరీ ట్రిప్లెట్‌గా మార్చి వ్రాయవచ్చన్నది ఇప్పటికే మీరు గ్రహించి ఉంటారు. కంప్యూటర్లో ఉండే గణిత, తర్క సర్క్యూట్లు (అరిథ్ మెటిక్, లాజిక్ యూనిట్ లేదా ఏఎల్‌యూ) మనం ఇచ్చే లెక్కలని వేసేది. ఇది బైనరీ పద్ధతిలోనే లెక్కలు వేసేది. ముందుగా కూడితలూ, తీసివేతలూ ఎలాచేస్తారన్నది చూద్దాం.

దశాంశ పద్ధతి	ఆక్టల్ పద్ధతి	బైనరీ పద్ధతి		
49	26	0	0	1 0
+ 53	+ 74	+0	+1	+ 1
----------	--------	-------	-------	--------
102	122	0	1	11
----------	--------	-------	-------	--------

ఈ బైనరీ పద్ధతిని మనం బాగా అర్థం చేసుకోవాలి. మరో ఉదాహరణ చూద్దాం. 110, 111 అన్న రెండు బైనరీ సంఖ్యలని ఎలా కూడాలో చూద్దాం.

```
   110
 + 111
 ----------
  11 01   -------  1 + 0  = 1
   |  |
   |  |______   1 + 1 = 10 . 0 వ్రాసి, 1 ని పక్క స్థానానికి తీసికెళ్ళాలి
 1+ 1 +1  =  10+ 1  = 11  అన్నమాట
```

బైనరీ పద్ధతి కూడికలన్నీ ఇలాగే జరుగుతాయి. పక్కని స్థానం నించి అప్పు తీసుకోవడం, లేదా మిగులు దాసుంచుకోవడం అన్నవాటిని భరించడానికి గణితశాస్త్రి సర్క్యూట్లలో బారో,క్యారీ (అప్పు తెచ్చుకో లేదా పక్కకి మోసుకెళ్ళు) జనరేటర్ అనేది ఒకటి ఉంటుంది. దీనినే ఫ్లిప్‌ఫ్లాప్ అని కూడా అంటారు. దీనిలో, పక్కస్థానానికి పంపాల్సిన బిట్ లేదా పక్కస్థానంనించీ అరువు తెచ్చుకున్న దానిని దాచుకుంటుంది. దీనికే ఓవర్ ఫ్లో బిట్ అని మరో పేరుంది. ఇపుడు ఒక బైనరీ తీసివేత ఎలా జరుగుతుందో చూద్దాం.

```
                  1 0 1
                  0 1 0
                  --------
                  0 1 1   ----------   1 - 0  = 1
                   |  |
1 ఇదివరకే అప్పిచ్చేశాం. మిగిలిందెల్లా --------  0 లోంచి  1 పోదు. పక్కనించీ
 0, 0 - 0  = 0                          1 అప్పు చేస్తే 10 - 1 = 1
```

లోటునీ, ఖర్చుల పోటునీ తట్టుకోడానికి మనం అరువెలా తెచ్చుకుంటున్నామో, అలాగే చిన్న మొత్తం నించీ పెద్ద మొత్తాన్ని తీసేయడం చేయాల్సి వచ్చినపుడు కంప్యూటర్ వాడే విధానాన్ని కాంప్లిమెంట్ పద్ధతి అంటారు. ఉదాహరణకి 27నించి 35ని తీసేయాలని అనుకుందాం. మామూలుగా మనం పెద్ద మొత్తాన్నించి చిన్న మొత్తాన్ని తీసేసి ముందు - (మైనస్) గుర్తు ఉంచేస్తాం కద. అంటే, 27 - 35 = - 8 ఔతుంది. ఇదే కాంప్లిమెంట్ పద్ధతిలో ఇలా జరుగుతుంది. పైన చెప్పుకున్న బైనరీ పద్ధతిలో లాగా చేస్తే, 2 7 - 3 5 = 9 2 ఔతుంది. అంటే, 7లో 5 తీసేస్తే 2 వచ్చింది. 2లో 3 పోదు కాబట్టి బైటనించి 1 అప్పు తెస్తే, 2 కాస్తా 12 ఔతుంది. ఇపుడు 12 లోంచి 3 సులభంగా పోయి 9 వచ్చి మొత్తం 92 ఔతుంది. ఇపుడు 92కి దగ్గరైన 10 గుణిజం అంటే 100 నించీ 92 ని తీసేస్తే 8 వస్తుంది. అసలు జరిగింది చిన్న మొత్తం నించి పెద్ద మొత్తం తీసివేత కాబట్టి - గుర్తు జోడించాలి. అంటే సమాధానం -8 అయ్యింది.

క్లుప్తంగా చెప్పాలంటే, 8కి 10 కాంప్లిమెంట్ 10 - 8 = 2. ఇదే బైనరీ పద్ధతిలో 2 కాంప్లిమెంట్ వాడతారు. అంటే, 101 అనే బైనరీ సంఖ్య కాంప్లిమెంట్ ఎలా వస్తుందీ అంటే, 101లో 0 కి 1, 1కి 0 వ్రాయాలి. అంటే 101 కాస్తా 010 ఔతుంది. దీనికి 1 కూడాలి. అంటే 011 ఔతుంది. ఈ 011 అనేది 101కి 2 కాంప్లిమెంటు. ఇట్లే గుణకార, భాగహారాలు జరుగుతాయి. ఈ వేదాంతమే కంప్యూటర్ల శాస్త్రంగా పేరొందిన బూలియన్ ఆల్జీబ్రా. ఇందులోని తర్కం చాలా ఉపయుక్తంగా ఉండి, కంప్యూటర్లు అతివేగంగా పనిచేసేటందుకు తోడ్పడ్తుంది. ఈ తర్క సిద్ధాంతానికే స్విచ్చింగ్ థియరీ అని పేరు.

కంప్యూటర్లలో పదాల సంగతులు!

కంప్యూటర్లకీ అసలు అర్థమయ్యేదెల్లా సున్న(0), ఒకటి(1) మాత్రమే అన్నది మళ్ళీ గుర్తు చెయ్యక్కరలేదు. మనం ఎలా ఎన్ని అవస్థలు పడి ప్రోగ్రామింగ్ చేసినా, కంప్యూటర్ చివరికెలాగోలా తంటాలు పడి వాటన్నిటినీ 0,1ల సముదాయంగా మార్చుకుని మాత్రమే అర్థంచేసుకుంటుంది. కంప్యూటర్లలో పదాలు ఎలా దాయబడతాయి, వాడబడతాయి అన్న విషయాలు కాస్త తెలుసుకుందాం! సాధారణంగా కంప్యూటర్లలో పదం (WORD), బిట్(BIT), బైట్(BYTE) అన్నవి వాడబడతాయి. ఈ పదం అన్నది ఒక నిజమైన బైనరీ సంఖ్య కావచ్చు, లేదా బైనరీలో కోడ్ చేయబడ్డ సంఖ్య కావచ్చు. లేదంటే, అక్షరాల సముదాయమూ కావచ్చు. ఆ అక్షరాలు బైనరీ కోడెడ్ గా ఉండవచ్చు. ఒక్క మాటలో చెప్పాలంటే, ఒక కంప్యూటర్ పదం అనేది కొన్ని బిట్‌ల సమూహం. ఒక బిట్ అంటే, బైనరీ డిజిట్. ఈ పదాల నిడివి 8 బిట్లు. ఒక్కోసారి అది 8 బిట్ల సమూహమూ కావచ్చు. ఒక్కో 8 బిట్‌ల సమూహాన్ని ఒక బైట్ అంటారు. 8 బిట్లు కలిస్తే ఒక బైట్. ఒక బైట్‌లో ఒక అక్షరాన్ని దాయవచ్చు. ఉదాహరణకి, ఐ..బి.ఎం.-360 మెయిన్‌ఫ్రేంలలో మెమరీలోని ప్రాథమిక పదం(దీనినే బేసిక్

యూనిట్ అంటారు) నిడివి 8 బిట్‌లు. అంటే, ప్రతీ అడ్రసున్న స్థానంలో 8 బిట్ల డేటా దాయవచ్చు.

మెమరీ యూనిట్	ఎన్ని బైట్లు	నిడివి(బిట్లలో)
బైట్	1	8
హాఫ్‌వర్డ్(సగం పదం)	2	16
వర్డ్(పూర్తి పదం)	4	32
ఫుల్‌వర్డ్(రెండింతల పదం)	2	64

కంప్యూటర్లలో ఆజ్ఞలు!

కంప్యూటర్లలో ఆజ్ఞల సముదాయం(ఆంగ్లంలో ఇన్‌స్ట్రక్షన్ సెట్ అంటారు)కి ఓ విశిష్ఠ స్థానం ఉంది. ఎలాటి పని(ఆపరేషన్ అని ఆంగ్ల రూపాంతరం) చేయబడుతుందో, దాని ఆధారంగా, లేకుంటే ఆజ్ఞ పదాలలో అమరిక ఎలా ఉందో దాని ఆధారంగా - ఇలా పలురకాలుగా వర్గీకరించబడతాయి. ఐతే, ముఖ్యంగా వాడుకలో రెండు రకాల ఆజ్ఞల సముదాయాలు వాడుకలో ఉన్నాయి. ఇవే మొదటి రకం, రెండో రకం(ఆంగ్లంలో వీటిని వరసగా టైప్-వన్, టైప్-టు ఇన్‌స్ట్రక్షన్స్ అంటారు) ఆజ్ఞలు అని పిలువబడుతున్నాయి. మొదటి రకం ఆజ్ఞలు కింద చెప్పినట్టు సూచింపబడతాయి.

పనికోడ్	నిరర్థకం	తోక	చిరునామా
12బిట్లు	6బిట్లు	3బిట్లు	15బిట్లు

---------------- 36 బిట్లు ----------------

ఒక 36 బిట్ల ఆజ్ఞలో 12 బిట్లు పని(ఆంగ్లంలో ఆపరేషన్ అంటారు) కోడ్ ని సూచిస్తుంది. దీని తరువాతి 6 బిట్లు దేనికీ వాడకుండా నిరర్థకంగా ఉండిపోతాయి. అటు తరువాతి 18 బిట్లలో 3 బిట్లు తోక(లేదా టాగ్) గానూ, మిగతా 15 బిట్లు చిరునామా(లేదా అడ్రస్)కి వాడబడుతుంది. పనికోడ్ అన్నది మనం చేయబోయే

పని కూడికనా? తీసివేతనా? లేక భాగహారమా? - అన్న దాని గూర్చి చెబుతుంది. తోకబిట్లన్నవి ఇండెక్స్ రిజిస్టర్లనే వాటిని సూచిస్తుంది. చివరి 15 బిట్లూ కీలకమైనవే. ఇవి ముందు పనికోడ్ లో చెప్పిన పని ఏఏ మెమరీ స్థానాలలో ఉన్న పదాలని వాడాలో చెబుతాయి.

రెండో రకం ఆజ్ఞలో పైన చెప్పిన 36బిట్లూ కింద చూపించినట్లుగా సూచింపబడుతుంది.

పనికోడ్ 3బిట్లు	తగ్గింపు 15బిట్లు	తోక 3బిట్లు	చిరునామా 15బిట్లు

----------------- 36 బిట్లు -----------------

రెండో రకం ఆజ్ఞకి చెందిన ఒక 36 బిట్ల ఆజ్ఞ, పైన చెప్పుకున్నట్లు మొదటిరకం ఆజ్ఞలానే ఉంటుంది. ఎటొచ్చీ ఇందులో నిరర్థకమయ్యే బిట్లు ఉండవు. పనికోడ్ 3బిట్లూ, తర్వాత 15బిట్లు తగ్గింపు (లేదా డిక్రిమెంట్)ని సూచిస్తుంది. తదుపరి 3 బిట్లు తోకబిట్లూ, చివరి 15 బిట్లు మామూలు గానే చిరునామాగానూ వాడబడతాయి.

ఇదే గణితపు పనులు(లేదా అరిథ్ మెటిక్ ఆపరేషన్స్) అన్నవి స్థిర, అస్థిర(వీటినే ఆంగ్లంలో వరసగా ఫిక్సెడ్ పాయింట్, ఫ్లోట్‌పాయింట్) పద్ధతులలో సూచించడం జరుగుతుంది. స్థిర పద్ధతి అంటే, యథాతథంగా మనం అంకెలనీ, సంఖ్యలనీ మనం ఎలా వ్రాస్తామో అలానే చూపడం. అంటే, 36బిట్ల నిడివి పదంలో మొదటి బిట్ ప్లస్ లేదా మైనస్ గుర్తుని సూచిస్తుంది. మిగిలిన 35బిట్లూ సంఖ్య యొక్క పరిమాణాన్ని(మాగ్నిట్యూడ్) సూచిస్తుంది.

ఐతే, గణితపు పనులలో ఉత్పన్నమయ్యే భిన్నాలని సూచించడానికి ఈ పద్ధతి పనికిరాదు. దానికి పనికి వచ్చేదే రెండో పద్ధతైన అస్థిర పద్ధతి. దీనిలో 36బిట్లలో మొదటిది యథా ప్రకారం ప్లస్ లేదా మైనస్ అన్నది చెబుతుంది. మిగిలిన 35బిట్లలో కొంత ఘాతము(ఎక్స్పొనెంట్) ఏమిటన్నది చూపడానికి, చివరగా మిగిలి నది ఘాతాంకము(మల్టిప్లయర్) ఏమిటన్నది చూపడానికి వాడబడుతుంది.

కంప్యూటర్ సంఖ్యలనెలా సరిదిద్దుకుంటుంది?

మామూలుగా మనం వాడే గణిత శాస్త్రాన్నే కంప్యూటర్లలో కూడా వాడటం జరుగుతుందని మనకి తెలిసింది. మనం గణితంలో వచ్చే భిన్నాలని ఒక్కోసారి భిన్నాల రూపంలో కాకుండా, దశాంశ పద్ధతిలోకి మార్చి మరీ సూచిస్తాం. అలా సూచించేపుడు ఈ భిన్నాలు ఎంతకీ అంతం కాకుండా అలా ఉండిపోతాయి. అపుడు మనం ఏదో ఓ స్థానం దాకా ఉంచుకుని మిగతా వాటిని అలాగే వాడకుండా వదిలేస్తాం. అంటే ఓ భిన్నం దశాంశ పద్ధతిలో 96.924123567 అని ఐదనుకుందాం. మనం రెండు స్థానాలకి మామూలుగా సవరించుకునీ కేవలం 96.92 అనే సూచిస్తాం. ఎందుకంటే, తరువాతి స్థానాలకి ప్రాముఖ్యాన్ని ఇవ్వడంలేదు మనం. దీన్నే తెనుగులో కత్తిరింపు పద్ధతి అని పిలుచుకోవచ్చు. ఆంగ్లంలో దీన్ని ట్రంకేషన్ మెథడ్ అంటారు. అదే 96.12532121 అనేది ఉందనుకుందాం. దీనిని రెండు దశాంశ స్థానాలకు కుదించాలన్నపుడు, మూడో స్థానం(అంటే రెండు స్థానాలకి కుదించాలి కాబట్టి) లో ఉన్న అంకె 2 వ స్థానంలో ఉండే అంకెలో సగం కన్నా ఎక్కువుంటే, 2వ స్థానానికి 1 కూడతాం. అంటే 96.13 అవుతుంది. దీన్నే సవరించే పద్ధతి (ఆంగ్లంలో రౌండింగ్) అనంటారు. కింది ఉదాహరణలు చూస్తే మీకు అంతా బోధపడుతుంది.

సంఖ్య	తగ్గింపు / కత్తిరింపు పద్ధతి	సవరింపు పద్ధతి	సవరించిన / కత్తిరించిన స్థానాలు
46.125321	46.12	46.13	2
46.1243521	46.12	46.12	2
3.3479001	3.34	3.35	2
26.72359	26.7235	26.723	4

ఈ సవరణ అనేది మనం ఎన్ని స్థానాలదాకా చేస్తాం అన్నదానిపై ఆధారపడివుంటుంది. ఐతే, ఈ కత్తిరింపులూ, కుదింపులూ చేయడం వల్ల చేసే లెక్కలలో ఫలితాలు కొద్దిగా తేడా వచ్చే ప్రమాదం లేకపోలేదు. దీనికే కత్తిరింపు దోషం లేదా ట్రంకేషన్ ఎర్రర్ అని పేరు. సవరణ లేదా కత్తిరింపు - ఏది చేసినా వీలైనంత తక్కువ ట్రంకేషన్ ఎర్రర్ ఉండేలా చుసుకోవాలి. మనం చేత్తో చేసే విధంగానే ఇలా కత్తిరింపులూ, సవరణలూ కంప్యూటర్లలోనూ వాడబడ్తాయి. ఇట్లే, మనం గణితంలో వాడే నికరమైన

పద్ధతులు(ప్రెసిషన్ మెథడ్స్ అని ఆంగ్లం) కంప్యూటర్లలోనూ వాడతారు. అవే ఏక, ద్వినికర పద్ధతులు. వీటిని ఆంగ్లంలో వరసగా సింగిల్ ప్రెసిషన్, డబుల్ ప్రెసిషన్ అంటారు. సింగిల్ ప్రెసిషన్‌లో నిజ స్థిరసంఖ్య (REAL CONSTANT) ఒక పదాన్ని ఆక్రమిస్తుంది. ఘాతం 7 బిట్లని, ఘాతాంకం 24 బిట్లనీ ఆక్రమిస్తుంది. ఒక బిట్ ప్లస్ (లేదా మైనస్) గుర్తుని సూచిస్తుంది. అంటే మొత్తం 32 బిట్లు.

నిజ స్థిరసంఖ్యలు 7 స్థానాల దాకా దాయబడతాయి. డబుల్ ప్రెసిషన్‌లో ఇదే విలువ 7 స్థానాల దాకా కాకుండా, 15 లేదా 16 గణనీయ(SIGNIFICANT) స్థానాలలో దాయబడ్తాయి. ఒక అంకె గానీ, ఒక సంఖ్య గానీ ఘాతము, ఘాతాంకమూ లేకుండా మామూలుగా చూపాల్సివస్తే, 8 నించి 16 స్థానాలలో చూపవచ్చు. డబుల్ ప్రెసిషన్ పద్ధతిలో సింగిల్ ప్రెసిషన్ పద్ధతిలో దాయబడే దానికన్నా రెండితలు దాయవచ్చు. అంటే డబుల్ ప్రెసిషన్ పద్ధతిలో దాయబడ్డ సంఖ్య విలువ సింగిల్ ప్రెసిషన్ పద్ధతిలో దాయబడేదాని కన్నా మరింత ఖచ్చితంగా ఉంటుంది. ఈ డబుల్ ప్రెసిషన్, సింగిల్ ప్రెసిషన్ పద్ధతిలో దాయడానికి హెక్సాడెసిమల్ పద్ధతినే ఎక్కువగా వాడతారు.

ఎన్‌కోడింగ్, డీకోడింగ్

మామూలుగా మనం మాట్లాడేటపుడు ఎలాటి సంకేతాలనీ వాడకుండా తిన్నగా మనం వాడే భాషని వాడి మాట్లాడుతాం. ఎలాటి ఇబ్బందీ ఉండదు. అదే టెలిగ్రాఫిక్ భాష ఉందనుకోండి. అందులో వాడేది కేవలం గుర్తులు. వాటినే డాట్(.), డాష్(-) అని అంటారు. దీన్ని మోర్స్ (Morse) కనుగొన్నాడు కాబట్టే దీన్ని మోర్స్‌కోడ్(Morse Code) అంటారు. A,B,C,D లకు డాట్, డాష్‌లని వాడతారు. అంటే A,B,C,D లు కోడ్‌ల రూపంలో వాడతారన్నమాట. దీన్నే కోడింగ్(CODING) అంటారు. ఒక మెసేజిని ఇలా ఒక టెలిగ్రాఫ్ ఆఫీసు నించి మరో టెలిగ్రాఫ్ ఆఫీసుకి చేరుతుంది. ఆ మెసేజీని అందుకున్న తర్వాత మళ్ళీ A,B,C,D లుగా మారుస్తారు. దీన్నే డీ-కోడింగ్(DE-CODING) అంటారు.

అదే పద్ధతిని కంప్యూటర్ పనిచేసే విధానానికి అన్వయించుకుందాం. కంప్యూటర్ పనిచేయాలంటే, మనం ప్రోగ్రాం, దానికి సంబంధించిన డేటా అందించాలి. ప్రోగ్రాం మనం వ్రాసిన ప్రోగ్రామింగ్ భాషనించి 0,1 భాషలోకి మార్చుకుని మరీ పని చేస్తుంది అని మనకు ఇదివరకే తెలుసు. మనం ఇచ్చిన డేటా కూడా ఇలాగే 0,1 లుగా మారిన తర్వాతే కంప్యూటర్ తన పనిని తాను సక్రమంగా నిర్వహించుకుపోతుంది. ఆ డేటా ఆల్ఫా న్యూమరిక్ అయినా సరే. అచ్చంగా న్యూమరిక్ అయినా సరే. ఉదాహరణకి, మనం RAMANA అనే డేటాని ఇన్‌పుట్‌గా ఇచ్చాం అనుకుందాం. అది R,A,M,A,N,A - అంటే రమణ అని మనకి అర్థం ఔతుంది కానీ

మనం ఆ డేటాని అందించడానికి వాడే కీబోర్డుకి గానీ, కంప్యూటర్ కి గానీ తెలియదు. కీబోర్డునించి వచ్చిన ఎలాటి డేటాని(లేదా అక్షరం) ఐనా ఒక రకమైన కోడింగ్ రూపంలో విద్యుత్ స్పందనలుగా సర్క్యూట్ల ద్వారా కంప్యూటర్ ప్రాసెసర్ కి అందుతుంది. ఇలా ఇన్‌పుట్‌ని ప్రాసెసర్ కి అర్థం అయ్యేలా మన డేటాని మార్చే ప్రక్రియనే కోడింగ్ అంటారు. ఇది మనం ప్రోగ్రాంనీ, డేటానీ ఇన్‌పుట్ చేసిన సమయంలో జరుగుతుంది. ఇలా కోడ్ చేయబడిన డేటా ప్రాసెసింగ్ పూర్తి అయ్యాక ఫలితాల రూపంలో తిరిగి మనకి అర్థమయ్యే భాషలో మార్చి ఔట్‌పుట్‌గా ఇస్తుంది కంప్యూటర్ ప్రాసెసర్. దీన్నే డీ-కోడింగ్ అంటారు. ఈ డీ-కోడింగ్ అనేది ఔట్‌పుట్ తయారయ్యే సమయంలో జరుగుతుంది.

ఈ డేటాని నెట్‌వర్క్‌లలో పంపడానికీ, అందుకోవడానికీ అనేక పవర్ ఫుల్ ఎన్‌కోడింగ్ డీ-కోడింగ్ పద్ధతులున్నాయి. అలా డేటాని కోడింగ్ చేయడాన్ని ఎన్‌క్రిప్షన్(Encryption) అనీ డీకోడింగ్ చేయడాన్ని డిక్రిప్షన్(Decryption) అనీ అంటారు. అలా చేయడానికి ఎన్నో పద్ధతులున్నాయి.

0000000