MEDHA AI

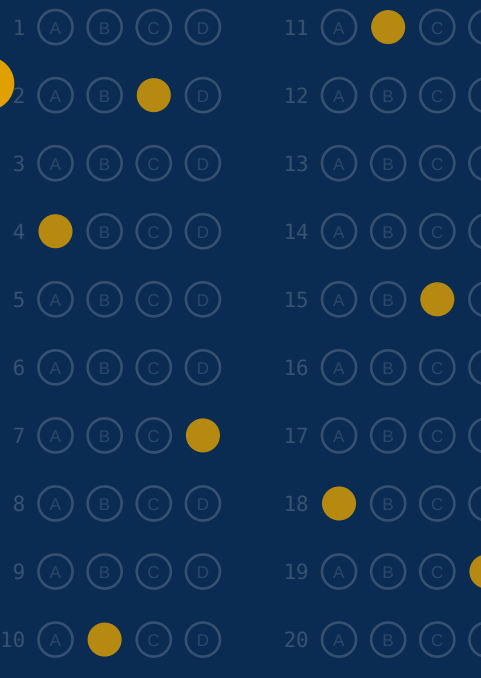
BASICS OF COMPUTERS

Computer basics: number systems, codes, logic, algorithms and spreadsheets

5 అంశాలు · జవాబులతో 18 ప్రాక్టీస్ ప్రశ్నలు · 2026



ఉచిత అధ్యాయం · app.medhaai.in లో మొత్తం కోర్సు



విషయ సూచిక

BASICS OF COMPUTERS

1 Computer basics: number systems, codes, logic, algorithms and spreadsheets 5 అంశాలు

1.1 అల్గారిథం, ఫ్లోచార్ట్: గుర్తులు, లక్షణాలు, ఉదాహరణలు 3

1.2 సంఖ్యా విధానాలు: బైనరీ, ఆక్టల్, హెక్సాడెసిమల్, దశాంశంలోకి, దశాంశం నుంచి మార్పిడి 7

1.3 స్ప్రెడ్‌షీట్లు: సెల్‌లు, రేంజ్‌లు, ఫార్ములాలు, సెల్ రిఫరెన్స్‌లు, ఫంక్షన్లు, ఎడిటింగ్, ప్రింటింగ్ 11

1.4 కోడ్‌లు (BCD, ASCII, EBCDIC, Unicode), బిట్‌లు, బైట్‌లు, వర్డ్‌లు, కంప్యూటర్ మెమరీ 16

1.5 బైనరీ అంకగణితం, పూరకాలు; లాజిక్ గేట్‌లు, తార్కిక ప్రక్రియలు 20

త్వరిత పునశ్చరణ 24

ప్రాక్టీస్ ప్రశ్నలు, జవాబులు 27

అధ్యాయం 1 · Basics of computers

Computer basics: number systems, codes, logic, algorithms and spreadsheets

5 అంశాలు

1.1 అల్గారిథం, ఫ్లోచార్ట్: గుర్తులు, లక్షణాలు, ఉదాహరణలు

అల్గారిథం (algorithm) అంటే ఒక సమస్యను పరిష్కరించడానికి పరిమిత సంఖ్యలో, అంచెలంచెలుగా, సందేహానికి తావులేని పద్ధతి; ఫ్లోచార్ట్ (flowchart) అవే సోపానాలను బాణాలతో కలిపిన ప్రామాణిక గుర్తులతో చిత్రంగా చూపుతుంది.

ముఖ్య వాస్తవాలు

- **అల్గారిథం:** ఇన్‌పుట్ (input)ను కావలసిన అవుట్‌పుట్ (output)గా మార్చే, స్పష్టంగా నిర్వచించిన సోపానాల పరిమిత క్రమం. ఈ పదం 9వ శతాబ్దపు పర్షియన్ గణిత శాస్త్రవేత్త **అల్-ఖ్వారిజ్మీ (al-Khwarizmi)** పేరు నుంచి వచ్చింది.
- **ఐదు లక్షణాలు** (క్నూత్ (Knuth)):
  - **పరిమితత్వం (Finiteness):** పరిమిత సంఖ్యలో సోపానాల తర్వాత అది ఆగిపోవాలి.
  - **నిర్దిష్టత (Definiteness):** ప్రతి సోపానం ఖచ్చితంగా, సందేహానికి తావు లేకుండా ఉండాలి.
  - **ఇన్‌పుట్: సున్నా లేదా అంతకంటే ఎక్కువ** ఇన్‌పుట్‌లు.
  - **అవుట్‌పుట్: కనీసం ఒక** అవుట్‌పుట్.
  - **ప్రభావశీలత (Effectiveness):** ప్రతి సోపానం ఖచ్చితంగా, పరిమిత సమయంలో (కాగితం, కలంతో) చేయగలిగేంత సరళంగా ఉండాలి.
- **అల్గారిథం రాసే విధానాలు:** సాధారణ భాషలో సోపానాలు, **సూడోకోడ్ (pseudocode)** (ప్రోగ్రామ్‌ను పోలిన ఆంగ్లం, కఠినమైన వ్యాకరణ నియమాలు ఉండవు), **ఫ్లోచార్ట్** (రేఖాచిత్రం).
- **ఫ్లోచార్ట్ గుర్తులు:**

గుర్తు	ఆకారం	ఉపయోగం
టెర్మినల్ (Terminal)	అండాకారం (oval) / గుండ్రటి మూలల దీర్ఘచతురస్రం	ప్రారంభం, ముగింపు
ఇన్‌పుట్/అవుట్‌పుట్	సమాంతర చతుర్భుజం (parallelogram)	చదవడం, ముద్రించడం
ప్రక్రియ (Process)	దీర్ఘచతురస్రం	లెక్కింపు, విలువ కేటాయింపు (assignment)

గుర్తు	ఆకారం	ఉపయోగం
నిర్ణయం (Decision)	వజ్రాకారం (diamond / rhombus)	అవును/కాదు లేదా సత్యం/అసత్యం పరీక్ష
ప్రవాహ రేఖ (Flow line)	బాణం	ప్రవాహ దిశ
కనెక్టర్ (Connector)	చిన్న వృత్తం	ఒకే పేజీలోని భాగాలను కలుపుతుంది
ఆఫ్-పేజ్ కనెక్టర్ (Off-page connector)	పంచభుజి (pentagon, హోమ్-ప్లేట్ ఆకారం), పాఠ్యపుస్తక సంప్రదాయం	మరో పేజీలో కొనసాగుతుంది
ముందే నిర్వచించిన ప్రక్రియ (Predefined process)	రెండు వైపులా రెండు నిలువు గీతలున్న దీర్ఘచతురస్రం	ఉప-ప్రోగ్రామ్ (sub-routine) పిలుపు

- **నిర్ణయ పెట్టె (decision box):** ఒక ప్రవేశం, రెండు (లేదా అంతకంటే ఎక్కువ) నిష్క్రమణలు. మిగిలిన పెట్టెలకు సాధారణంగా ఒకే నిష్క్రమణ ఉంటుంది.
- **మూడు నియంత్రణ నిర్మాణాలు (control structures)** (బోమ్–జకోపిని (Böhm–Jacopini), 1966): **క్రమం (sequence)**, **ఎంపిక (selection)** (if / if-else / case), **పునరావృతం (iteration)** (లూప్‌లు (loops): while, do-while/repeat-until, for). ఏ అల్గారిథాన్నైనా ఈ మూడింటితో నిర్మించవచ్చు.
- **ఫ్లోచార్ట్ సంప్రదాయాలు:** ప్రవాహం **పై నుంచి కిందికి, ఎడమ నుంచి కుడికి**; సరిగ్గా ఒక ప్రారంభం, కనీసం ఒక ముగింపు; ప్రవాహ రేఖలు ఒకదానినొకటి దాటకూడదు (కనెక్టర్లు వాడాలి).

ఉదాహరణ 1. "A, B చదువు" కోసం ఏ ఫ్లోచార్ట్ గుర్తు వాడతారు? సాధన. ఇన్‌పుట్, అవుట్‌పుట్‌లకు సమాంతర చతుర్భుజం వాడతారు. **సమాధానం: సమాంతర చతుర్భుజం**

ఉదాహరణ 2. ఒక అల్గారిథం ఎలాంటి ఇన్‌పుట్ తీసుకోకుండా "Hello" అని ఒకసారి ముద్రిస్తుంది. అది ఇంకా ఏ లక్షణాన్ని పాటిస్తుంది? సాధన. ఇన్‌పుట్ సున్నా కావచ్చు; అవుట్‌పుట్ కనీసం ఒకటి ఉండాలి. **సమాధానం: ఇది సరైన అల్గారిథమే (సున్నా ఇన్‌పుట్‌లు, ఒక అవుట్‌పుట్)**

1. అల్గారిథాన్ని సోపానాలుగా రాయడం.

ఉదాహరణ 3. రెండు సంఖ్యలలో పెద్దదాన్ని కనుక్కునే అల్గారిథం.

1. ప్రారంభం.
2. A, B చదువు.
3. A > B అయితే A ను ముద్రించు; లేకపోతే B ను ముద్రించు.
4. ముగింపు.

ఫ్లోచార్ట్: అండాకారం ప్రారంభం → సమాంతర చతుర్భుజం *A, B* చదువు → వజ్రాకారం *A > B?* → అవును: సమాంతర చతుర్భుజం *A* ను ముద్రించు; కాదు: సమాంతర చతుర్భుజం *B* ను ముద్రించు → రెండు బాణాలు కలుస్తాయి → అండాకారం ముగింపు.

2. మూడు సంఖ్యలలో పెద్దది (ఒకదానిలో ఒకటి ఉన్న ఎంపిక, nested selection).

1. ప్రారంభం. A, B, C చదువు.
2. A > B అయితే: A > C అయితే A ను ముద్రించు, లేకపోతే C ను ముద్రించు.
3. లేకపోతే: B > C అయితే B ను ముద్రించు, లేకపోతే C ను ముద్రించు.
4. ముగింపు.

ఈ ఫ్లోచార్ట్‌లో ఏ మార్గంలో వెళ్లినా సరిగ్గా **రెండు** నిర్ణయ పెట్టెలు వస్తాయి.

3. లూప్‌లు: మొత్తం, ఫ్యాక్టోరియల్.

ఉదాహరణ 4. N! (ఫ్యాక్టోరియల్) కోసం అల్గారిథం.

1. ప్రారంభం. N చదువు.
2. F ← 1, I ← 1.
3. I ≤ N గా ఉన్నంత వరకు: F ← F × I; I ← I + 1.
4. F ను ముద్రించు. ముగింపు.

N = 5 కి డ్రై రన్: F వరుసగా 1, 2, 6, 24, 120 విలువలు తీసుకుంటుంది. **సమాధానం: 120**

ఫ్లోచార్ట్: *I ≤ N?* అనే పరీక్ష వజ్రాకారంలో ఉంటుంది; "అవును" శాఖ *F ← F × I, I ← I + 1* అనే ప్రక్రియ పెట్టెకు వెళ్తుంది, అక్కడి నుంచి ఒక బాణం తిరిగి వజ్రాకారానికి వస్తుంది; "కాదు" శాఖ *F* ను ముద్రించుకు వెళ్తుంది.

4. లూప్ ఎన్నిసార్లు నడుస్తుందో లెక్కించడం.

ఉదాహరణ 5. "for I = 1 to 10 step 3" లూప్ లోపలి భాగం ఎన్నిసార్లు నడుస్తుంది? సాధన. I = 1, 4, 7, 10; తర్వాతి విలువ 13, ఇది 10 కంటే ఎక్కువ. **సమాధానం: 4 సార్లు**

5. ముందు పరీక్షించే, తర్వాత పరీక్షించే లూప్‌లు.

లూప్	పరీక్ష	కనీసం ఎన్నిసార్లు నడుస్తుంది
while / for	లోపలి భాగానికి ముందు (ప్రవేశ నియంత్రిత, entry-controlled)	0
do-while / repeat-until	లోపలి భాగం తర్వాత (నిష్క్రమణ నియంత్రిత, exit-controlled)	1

6. రెండు విలువలను పరస్పరం మార్చడం (swapping).

ఉదాహరణ 6. A = 5, B = 3. సోపానాలు: A ← A + B; B ← A − B; A ← A − B. చివరి విలువలు? సాధన. A = 8; B = 8 − 3 = 5; A = 8 − 5 = 3. **సమాధానం: A = 3, B = 5** (మూడో చరరాశి లేకుండానే మారాయి). సాధారణ పద్ధతిలో ఒక తాత్కాలిక చరరాశి (variable) వాడతారు: T ← A; A ← B; B ← T.

7. డ్రై రన్ (ట్రేస్ పట్టిక). నమూనా విలువలతో అల్గారిథాన్ని చేతితో అమలు చేస్తూ, ప్రతి సోపానం తర్వాత ప్రతి చరరాశి విలువను రాయడాన్ని **డ్రై రన్ (dry run)** లేదా **డెస్క్ చెక్ (desk check)** అంటారు. కోడ్ రాయకముందే తార్కిక దోషాలను ఇది పట్టుకుంటుంది.

లోతైన విశ్లేషణ

1. గ.సా.భా. (HCF / GCD) కోసం యూక్లిడ్ అల్గారిథం.

1. A, B చదువు.
2. B ≠ 0 గా ఉన్నంత వరకు: R ← A mod B; A ← B; B ← R.
3. A ను ముద్రించు.

ఉదాహరణ 7. A = 48, B = 18 కి ట్రేస్.

విడత	A	B	R = A mod B
1	48	18	12
2	18	12	6
3	12	6	0

3వ విడత తర్వాత A = 6, B = 0. **సమాధానం: గ.సా.భా. 6, లూప్ 3 విడతల తర్వాత**

2. అంకెల మొత్తం.

ఉదాహరణ 8. S ← 0; N > 0 గా ఉన్నంత వరకు: S ← S + (N mod 10); N ← N div 10. N = 4729 కి అవుట్‌పుట్? సాధన. వరుసగా తీసుకున్న అంకెలు 9, 2, 7, 4. S = 9 + 2 + 7 + 4. **సమాధానం: 22**

3. శోధన, క్రమబద్ధీకరణ (searching and sorting) (డిగ్రీ స్థాయిలో సాధారణంగా ఇచ్చే ఉదాహరణలు).

పద్ధతి	క్రమబద్ధీకరించిన డేటా అవసరమా?	చెత్త సందర్భంలో (worst case) పోలికలు
రేఖీయ శోధన (Linear search)	అవసరం లేదు	n
బైనరీ శోధన (Binary search)	**అవసరం**	$\lfloor \log_2 n \rfloor + 1$ (సుమారు $\log_2 n$)
బబుల్ సార్ట్ (Bubble sort)	—	$n(n-1)/2$, $n-1$ విడతల్లో

ఉదాహరణ 9. 1,000 అంశాలున్న క్రమబద్ధీకరించిన జాబితాలో బైనరీ శోధనకు గరిష్ఠంగా ఎన్ని పోలికలు? (a) 9 (b) 10 (c) 100 (d) 500 సాధన. ప్రతి పోలిక జాబితాను సగం చేస్తుంది; చెత్త సందర్భం = $\lfloor \log_2 1000 \rfloor + 1 = 9 + 1$ (ఎందుకంటే $2^9 = 512 \le 1{,}000 < 1{,}024 = 2^{10}$). **సమాధానం: (b) 10**

ఉదాహరణ 10. 6 సంఖ్యలపై బబుల్ సార్ట్ (ప్రాథమిక రూపం): ఎన్ని విడతలు, ఎన్ని పోలికలు? సాధన. విడతలు = 6 − 1 = 5. పోలికలు = 6 × 5 ÷ 2. **సమాధానం: 5 విడతలు, 15 పోలికలు**

4. ఫ్లోచార్ట్‌ల ప్రయోజనాలు, పరిమితులు.

- ప్రయోజనాలు: తర్కాన్ని స్పష్టంగా చూపుతాయి, సులభంగా వివరించవచ్చు, దోషాల సవరణ (debugging), డాక్యుమెంటేషన్‌లో ఉపయోగపడతాయి.
- పరిమితులు: పెద్ద ప్రోగ్రామ్‌లకు శ్రమతో కూడినవి, మార్చడం కష్టం (మళ్లీ గీయాలి), వివరాలను చూపడానికి ప్రామాణిక పద్ధతి లేదు; సూడోకోడ్‌ను మార్చడం సులభం.

5. అల్గారిథం సామర్థ్యం (efficiency). సరైన రెండు అల్గారిథాలు వాడే **సమయం** (సోపానాల సంఖ్య), **స్థలం** (మెమరీ (memory)) వేర్వేరుగా ఉండవచ్చు. బిగ్-O సంకేతం (Big-O notation) వృద్ధి రేటును చూపుతుంది: రేఖీయ శోధన $O(n)$, బైనరీ శోధన $O(\log n)$, బబుల్ సార్ట్ $O(n^2)$.

తరచూ చేసే పొరపాట్లు

- ✗ "అల్గారిథానికి కనీసం ఒక ఇన్‌పుట్ ఉండాలి." ✓ **సున్నా లేదా అంతకంటే ఎక్కువ** ఇన్‌పుట్‌లు; **కనీసం ఒక** అవుట్‌పుట్.
- ✗ "నిర్ణయాలకు దీర్ఘచతురస్రం వాడతారు." ✓ **వజ్రాకారం**; దీర్ఘచతురస్రం **ప్రక్రియ** కోసం.
- ✗ "ఇన్‌పుట్/అవుట్‌పుట్‌కు అలల అడుగు భాగం ఉన్న దీర్ఘచతురస్రం వాడతారు." ✓ **సమాంతర చతుర్భుజం**; అలల అడుగు భాగం ఉన్నది **డాక్యుమెంట్** గుర్తు.
- ✗ "ప్రారంభం/ముగింపు వృత్తం." ✓ **అండాకారం (టెర్మినల్)**; చిన్న వృత్తం **కనెక్టర్**.
- ✗ "బైనరీ శోధన ఏ జాబితాపైనైనా పనిచేస్తుంది." ✓ జాబితా **క్రమబద్ధీకరించి** ఉండాలి.
- ✗ "while లూప్ ఎప్పుడూ కనీసం ఒకసారి నడుస్తుంది." ✓ అది **సున్నా** సార్లు నడవవచ్చు; **do-while/repeat-until** కనీసం ఒకసారి నడుస్తుంది.
- ✗ "పరిమితత్వం అంటే ఇన్‌పుట్ పరిమితం అని." ✓ అల్గారిథం పరిమిత సంఖ్యలో సోపానాల తర్వాత **ఆగిపోతుంది** అని అర్థం.

1.2 సంఖ్యా విధానాలు: బైనరీ, ఆక్టల్, హెక్సాడెసిమల్, దశాంశంలోకి, దశాంశం నుంచి మార్పిడి

ఒక సంఖ్యా విధానాన్ని దాని ఆధారం (base) లేదా రాడిక్స్ (radix) నిర్ణయిస్తుంది: కంప్యూటర్లు బైనరీ (binary) (ఆధారం 2)లో పనిచేస్తాయి; ఆక్టల్ (octal) (ఆధారం 8), హెక్సాడెసిమల్ (hexadecimal) (ఆధారం 16) బైనరీని సంక్షిప్తంగా రాసే పద్ధతులు. ఏ ఆధారంలోని సంఖ్యనైనా అంకె × స్థాన విలువను కలపడం ద్వారా దశాంశంలోకి మార్చవచ్చు; దశాంశాన్ని ఏ ఆధారంలోకైనా పునరావృత భాగహారం (repeated division) (పూర్ణ భాగం), పునరావృత గుణకారం (repeated multiplication) (భిన్న భాగం) ద్వారా మార్చవచ్చు.

ముఖ్య వాస్తవాలు

- **ఆధారం (రాడిక్స్)** = విభిన్న అంకెల (digits) సంఖ్య. అంకెలు 0 నుంచి ఆధారం − 1 వరకు ఉంటాయి.

విధానం	ఆధారం	అంకెలు	ఉదాహరణ
బైనరీ	2	0, 1	101101_2
ఆక్టల్	8	0 నుంచి 7	55_8
దశాంశం (decimal)	10	0 నుంచి 9	45_{10}
హెక్సాడెసిమల్	16	0 నుంచి 9, A నుంచి F	$2D_{16}$

- **హెక్స్ అక్షరాలు:** A = 10, B = 11, C = 12, D = 13, E = 14, F = 15.
- **స్థాన విలువ (positional value):** ప్రతి స్థానం ఆధారం యొక్క ఒక ఘాతం. బైనరీలో కుడి నుంచి: 1, 2, 4, 8, 16, 32, 64, 128, ...; బిందువు తర్వాత: ½, ¼, ⅛, ...
- **ఏ ఆధారం → దశాంశం:** ప్రతి అంకెను దాని స్థాన విలువతో గుణించి కలపండి.
- **దశాంశం (పూర్ణ భాగం) → ఆధారం b:** b తో పదే పదే భాగించండి; **శేషాలను (remainders) కింది నుంచి పైకి** చదవండి.
- **దశాంశం (భిన్న భాగం) → ఆధారం b:** b తో పదే పదే గుణించండి; **పూర్ణాంక భాగాలను పై నుంచి కిందికి** చదవండి.
- **బైనరీ ↔ ఆక్టల్: 3 బిట్‌ల (bits)** గుంపులు ($8 = 2^3$). **బైనరీ ↔ హెక్స్: 4 బిట్‌ల** గుంపులు ($16 = 2^4$). బిందువు నుంచి బయటికి గుంపులు చేయండి; ఖాళీ స్థానాలను సున్నాలతో నింపండి.
- **ఆక్టల్ ↔ హెక్స్:** బైనరీ ద్వారా మార్చండి.
- **ఎంఎస్‌బీ (MSB)** = అత్యంత ప్రాముఖ్యం గల (ఎడమ చివరి) బిట్; **ఎల్ఎస్‌బీ (LSB)** = అత్యల్ప ప్రాముఖ్యం గల (కుడి చివరి) బిట్.
- **n బిట్‌లు 2^n** అమరికలను ఇస్తాయి; సంజ్ఞ లేని (unsigned) అతిపెద్ద విలువ **$2^n - 1$** (8 బిట్‌లు: 0 నుంచి 255).
- ఉపయోగపడే జతలు: $2^{10} = 1{,}024$; $255 = 11111111_2 = 377_8 = FF_{16}$.

ఉదాహరణ 1. 45_{10} ను బైనరీలోకి మార్చండి. *సాధన.* 45 ÷ 2 = 22 శేషం 1; 22 ÷ 2 = 11 శేషం 0; 11 ÷ 2 = 5 శేషం 1; 5 ÷ 2 = 2 శేషం 1; 2 ÷ 2 = 1 శేషం 0; 1 ÷ 2 = 0 శేషం 1. కింది నుంచి పైకి చదవండి. **సమాధానం: 101101_2**

ఉదాహరణ 2. 11010_2 ను దశాంశంలోకి మార్చండి. *సాధన.* 16 + 8 + 0 + 2 + 0. **సమాధానం: 26**

1. దశాంశం నుంచి ఆక్టల్, హెక్సాడెసిమల్‌లోకి.

ఉదాహరణ 3. 156_{10} ను ఆక్టల్, హెక్సాడెసిమల్‌లోకి మార్చండి. *సాధన.* ఆక్టల్: 156 ÷ 8 = 19 శేషం 4; 19 ÷ 8 = 2 శేషం 3; 2 ÷ 8 = 0 శేషం 2 → **234_8**. హెక్స్: 156 ÷ 16 = 9 శేషం 12 (C); 9 ÷ 16 = 0 శేషం 9 → **$9C_{16}$**. సరిచూడటం: 2 × 64 + 3 × 8 + 4 = 156; 9 × 16 + 12 = 156. **సమాధానం: 234_8, $9C_{16}$**

2. ఆక్టల్, హెక్సాడెసిమల్ నుంచి దశాంశంలోకి.

ఉదాహరణ 4. 754_8, $3F7_{16}$ లను దశాంశంలోకి మార్చండి. *సాధన.* 7 × 64 + 5 × 8 + 4 = 448 + 40 + 4 = 492. 3 × 256 + 15 × 16 + 7 = 768 + 240 + 7 = 1,015. **సమాధానం: 492, 1,015**

3. గుంపులుగా విభజించి బైనరీ నుంచి ఆక్టల్, హెక్స్‌లోకి.

ఉదాహరణ 5. 1101011_2 ను ఆక్టల్, హెక్సాడెసిమల్, దశాంశాలలో రాయండి. *సాధన.* కుడి నుంచి 3 బిట్‌ల గుంపులు: 1 | 101 | 011 → **153_8**. 4 బిట్‌ల గుంపులు: 110 | 1011 → 6 | B → **$6B_{16}$**. దశాంశం: 64 + 32 + 8 + 2 + 1 = **107**.

ఉదాహరణ 6. $A5_{16}$ ను ఆక్టల్‌లోకి మార్చండి. (a) 245 (b) 165 (c) 255 (d) 225 *సాధన.* A5 = 1010 0101 → మూడేసి బిట్‌లుగా మళ్ళీ గుంపులు: 10 | 100 | 101 → 2 | 4 | 5. **సమాధానం: (a) 245_8** ($A5_{16} = 245_8$ = దశాంశంలో 165; (b) దశాంశ విలువ, ఇది ఉచ్చు).

4. భిన్నాలు (fractions).

ఉదాహరణ 7. 0.625_{10} ను బైనరీలోకి మార్చండి. *సాధన.* 0.625 × 2 = 1.25 → 1; 0.25 × 2 = 0.5 → 0; 0.5 × 2 = 1.0 → 1. పై నుంచి కిందికి చదవండి. **సమాధానం: 0.101_2**

ఉదాహరణ 8. 25.375_{10} ను బైనరీలోకి మార్చండి. *సాధన.* 25 = 11001; 0.375 × 2 = 0.75 → 0; 0.75 × 2 = 1.5 → 1; 0.5 × 2 = 1.0 → 1, కాబట్టి 0.011. **సమాధానం: 11001.011_2**

ఉదాహరణ 9. 1101.11_2, $1A.8_{16}$ లను దశాంశంలోకి మార్చండి. *సాధన.* 8 + 4 + 1 + ½ + ¼ = 13.75. 1 × 16 + 10 + 8/16 = 26.5. **సమాధానం: 13.75, 26.5**

పద్ధతుల సారాంశం

నుంచి → లోకి	పద్ధతి
ఆధారం b → దశాంశం	Σ అంకె × b^స్థానం
దశాంశం → ఆధారం b	÷ b (పూర్ణ భాగం), × b (భిన్న భాగం)
బైనరీ → ఆక్టల్ / హెక్స్	3 / 4 బిట్‌ల గుంపులు

నుంచి → లోకి	పద్ధతి
ఆక్టల్ ↔ హెక్స్	బైనరీ ద్వారా

లోతైన విశ్లేషణ

- **ప్రతి దశాంశ భిన్నం బైనరీలో అంతం కాదు.** $0.1_{10} = 0.000110011\ldots_2$ (0011 అనే భాగం పునరావృతం అవుతుంది); అందుకే చాలా ప్రోగ్రామ్‌లలో 0.1 + 0.2 సరిగ్గా 0.3 కాదు. ఒక భిన్నం (కనిష్ఠ రూపంలో) హారం 2 యొక్క ఘాతం అయితేనే అది బైనరీలో అంతమవుతుంది.
- **దశాంశ సంఖ్య N కు కావలసిన బిట్‌లు:** $\lfloor \log_2 N \rfloor + 1$. 100 కు 7 బిట్‌లు (1100100); 1,000 కు 10 బిట్‌లు.
- **అంకెల చెల్లుబాటు:** ప్రతి అంకె b కంటే తక్కువగా ఉంటేనే ఆ సంఖ్య ఆధారం b లో చెల్లుతుంది. "789" ఆక్టల్ సంఖ్య కాదు; "1234" కు ఆధారం 5 లేదా అంతకంటే ఎక్కువ కావాలి.
- **ఆక్టల్, హెక్స్ ఎందుకు?** పొడవైన బిట్ వరుసలను ఇవి సంక్షిప్తం చేస్తాయి: ఒక హెక్స్ అంకె = 4 బిట్‌లు, రెండు హెక్స్ అంకెలు = ఒక బైట్ (byte). మెమరీ (memory) చిరునామాలు, రంగు కోడ్‌లు (#FF0000), మెషిన్ కోడ్ జాబితాలు హెక్స్‌ను వాడతాయి.
- **జరపడం (shifting):** బైనరీ సంఖ్య చివర 0 చేరిస్తే అది రెట్టింపు అవుతుంది (1011 → 10110, 11 → 22); చివరి బిట్‌ను తొలగిస్తే సగం అవుతుంది (పూర్ణ భాగం మాత్రమే).

ఉదాహరణ 10. ఆధారం b లో 121 దశాంశంలో 49 కి సమానమైతే, b ను కనుగొనండి. (a) 5 (b) 6 (c) 7 (d) 8 సాధన. $b^2 + 2b + 1 = 49$, కాబట్టి $(b + 1)^2 = 49$, $b = 6$. **సమాధానం: (b) 6**

ఉదాహరణ 11. $2F_{16}$ ను ఆక్టల్‌లోకి మార్చండి. సాధన. 2F = 0010 1111 = 101 111 → 57_8. సరిచూడటం: $2 \times 16 + 15 = 47 = 5 \times 8 + 7$. **సమాధానం: 57_8**

ఉదాహరణ 12. ఆరోహణ క్రమంలో అమర్చండి: 110011_2, 62_8, 33_{16}. సాధన. $110011_2 = 51$; $62_8 = 50$; $33_{16} = 51$. **సమాధానం: $62_8 < 110011_2 = 33_{16}$**

తరచూ చేసే పొరపాట్లు

- ✗ "శేషాలను పై నుంచి కిందికి చదవాలి." ✓ పూర్ణ సంఖ్యలకు **కింది నుంచి పైకి** చదవాలి (చివరి శేషమే ఎంఎస్‌బీ).
- ✗ "భిన్న భాగపు అంకెలను కింది నుంచి పైకి చదవాలి." ✓ భిన్నాలకు పూర్ణాంక భాగాలను **పై నుంచి కిందికి** చదవాలి.
- ✗ "బిట్‌లను ఎడమ నుంచి గుంపులు చేయాలి." ✓ **బైనరీ బిందువు నుంచి బయటికి** (పూర్ణ భాగానికి కుడి నుంచి ఎడమకు) గుంపులు చేసి, సున్నాలతో నింపాలి.
- ✗ "ఆక్టల్ 0 నుంచి 8 అంకెలను వాడుతుంది." ✓ ఆక్టల్ **0 నుంచి 7** వాడుతుంది; హెక్స్ 0 నుంచి 9, **A నుంచి F** వాడుతుంది.
- ✗ "హెక్స్ F = 16." ✓ **F = 15**; 10_{16} = 16.
- ✗ "8 బిట్‌లు 256 వరకు నిల్వ చేయగలవు." ✓ 8 బిట్‌లు **256 అమరికలను** ఇస్తాయి, కాబట్టి సంజ్ఞ లేని అతిపెద్ద విలువ **255**.
- ✗ "ప్రతి దశాంశ భిన్నం ఖచ్చితంగా మారుతుంది." ✓ 0.1_{10} బైనరీలో **పునరావృతం** అవుతుంది.

1.3 స్ప్రెడ్‌షీట్లు: సెల్‌లు, రేంజ్‌లు, ఫార్ములాలు, సెల్ రిఫరెన్స్‌లు, ఫంక్షన్లు, ఎడిటింగ్, ప్రింటింగ్

స్ప్రెడ్‌షీట్ (spreadsheet) అంటే అడ్డు వరుసలు (rows), నిలువు వరుసలు (columns) ఉన్న గ్రిడ్; దాని సెల్‌ల (cells)లో సంఖ్యలు, అక్షరాలు (text) లేదా ఫార్ములాలు (formulas) ఉంటాయి; ఒక ఫార్ములా వాడే ఏ విలువ మారినా అది తనంతట తానే మళ్లీ లెక్కిస్తుంది.

ముఖ్య వాస్తవాలు

- **స్ప్రెడ్‌షీట్ ప్రోగ్రామ్‌లు:** ఎంఎస్ ఎక్సెల్ (MS Excel), లిబ్రేఆఫీస్ కాల్క్ (LibreOffice Calc), గూగుల్ షీట్స్ (Google Sheets); పీసీ కోసం వచ్చిన మొదటి స్ప్రెడ్‌షీట్ **విజికాల్క్ (VisiCalc) (1979)**, దాని తర్వాత **లోటస్ 1-2-3 (Lotus)**.
- **వర్క్‌బుక్ (workbook)** = ఫైల్; **వర్క్‌షీట్ (worksheet)** (షీట్) = దాని లోపలి ఒక గ్రిడ్. ఎక్సెల్ ఫైల్ రకాలు: **.xlsx** (2007 నుంచి), **.xls** (పాతది), **.xlsm** (మ్యాక్రోలతో); కాల్క్ **.ods** వాడుతుంది; **.csv** అంటే కామాలతో వేరుచేసిన సాధారణ టెక్స్ట్.
- **అడ్డు వరుసలకు** సంఖ్యలు (1, 2, 3 ...); **నిలువు వరుసలకు** అక్షరాలు (A ... Z, AA ... XFD).
- **ఎక్సెల్ 2007 నుంచి:** ఒక్కో షీట్‌లో **1,048,576 అడ్డు వరుసలు × 16,384 నిలువు వరుసలు**; చివరి నిలువు వరుస **XFD**. (ఎక్సెల్ 2003: 65,536 అడ్డు వరుసలు × 256 నిలువు వరుసలు, చివరి నిలువు వరుస IV.)

- **సెల్:** ఒక అడ్డు వరుస, ఒక నిలువు వరుస కలిసే చోటు; దాని **అడ్రస్** = నిలువు వరుస అక్షరం + అడ్డు వరుస సంఖ్య, ఉదా. **C5**.
- **యాక్టివ్ సెల్ (active cell):** మందమైన అంచుతో కనిపించే, ఎంపిక చేసిన సెల్; దాని అడ్రస్ **నేమ్ బాక్స్ (Name Box)**లో, దానిలోని విషయం **ఫార్ములా బార్ (Formula Bar)**లో కనిపిస్తాయి.
- **రేంజ్ (range):** కోలన్‌తో రాసే దీర్ఘచతురస్రాకార సెల్‌ల సమూహం, ఉదా. **A1:C4** (3 నిలువు వరుసలు × 4 అడ్డు వరుసలు = **12 సెల్‌లు**).
- **ఫార్ములా:** **=** తో మొదలవుతుంది (లోటస్ +/@ వాడేది), ఉదా. =A1+B1. **ఫంక్షన్ (function)** అంటే ముందే తయారుచేసి ఉంచిన ఫార్ములా, ఉదా. =SUM(A1:A10).
- **సెల్ రిఫరెన్స్‌లు:**
  - **రిలేటివ్ (relative)** A1: కాపీ చేసినప్పుడు మారుతుంది.
  - **అబ్సల్యూట్ (absolute)** A1: ఎప్పుడూ మారదు.
  - **మిక్స్‌డ్ (mixed)** $A1 (నిలువు వరుస స్థిరం) లేదా A$1 (అడ్డు వరుస స్థిరం).
  - **F4** నొక్కితే A1 → A1 → A$1 → $A1 → A1 గా వరుసగా మారుతుంది.
- **డిఫాల్ట్ అమరిక (alignment):** సంఖ్యలు, తేదీలు **కుడివైపు**, టెక్స్ట్ **ఎడమవైపు**.
- **ఫిల్ హ్యాండిల్ (fill handle):** ఎంపిక చేసిన భాగం కుడి-దిగువ మూలలోని చిన్న చతురస్రం; దాన్ని లాగితే కాపీ అవుతుంది లేదా శ్రేణి **ఆటోఫిల్ (AutoFill)** అవుతుంది (Mon, Tue ...; 1, 2, 3 ...).
- **చార్ట్, సార్ట్, ఫిల్టర్, పివట్ టేబుల్, కండిషనల్ ఫార్మాటింగ్** ప్రామాణిక సౌకర్యాలు.

ఉదాహరణ 1. B2:E6 రేంజ్‌లో ఎన్ని సెల్‌లు ఉన్నాయి? సాధన. B నుంచి E వరకు నిలువు వరుసలు = 4; 2 నుంచి 6 వరకు అడ్డు వరుసలు = 5. 4 × 5. **సమాధానం: 20**

1. నమోదు చేయడం, సవరించడం.

- **F2** యాక్టివ్ సెల్‌ను సవరిస్తుంది; **Esc** నమోదును రద్దు చేస్తుంది; **Enter** నిర్ధారించి కిందికి కదులుతుంది; **Tab** కుడివైపుకు కదులుతుంది.
- **Ctrl+Z** అన్‌డూ, **Ctrl+Y** రీడూ, **Ctrl+C / X / V** కాపీ, కట్, పేస్ట్; **Ctrl+F** వెతకడం, **Ctrl+H** మార్చడం (replace).
- **పేస్ట్ స్పెషల్ (Paste Special)** విలువలు, ఫార్మాట్‌లు లేదా ఫార్ములాలు మాత్రమే పేస్ట్ చేస్తుంది, లేదా అడ్డు, నిలువు వరుసలను తారుమారు (transpose) చేస్తుంది.
- **Ctrl+;** ఈ రోజు తేదీని చేరుస్తుంది; **Ctrl+Home** A1 కు వెళ్తుంది; **Alt+Enter** సెల్ లోపలే కొత్త లైన్ మొదలుపెడుతుంది.
- అడ్డు వరుసలు, నిలువు వరుసలు, సెల్‌లను **చేర్చడం / తొలగించడం (Insert / Delete)**; **క్లియర్ (Clear)** విషయాన్ని తీసేస్తుంది కానీ సెల్ అలాగే ఉంటుంది; **డిలీట్** సెల్‌నే తీసేసి మిగిలినవాటిని జరుపుతుంది.
- **మెర్జ్ & సెంటర్ (Merge & Center)** సెల్‌లను కలుపుతుంది; **ర్యాప్ టెక్స్ట్ (Wrap Text)** పొడవైన టెక్స్ట్‌ను సెల్‌లో ఇమిడేలా చేస్తుంది; **ఫ్రీజ్ పేన్స్ (Freeze Panes)** స్క్రోల్ చేస్తున్నప్పుడు శీర్షికలు కనిపించేలా ఉంచుతుంది.

- సెల్‌లో **####**: సంఖ్యను చూపడానికి నిలువు వరుస **చాలా ఇరుకుగా** ఉంది (వెడల్పు పెంచండి).

2. అంకగణిత ఆపరేటర్లు, ప్రాధాన్య క్రమం (ఎక్సెల్).

క్రమం	ఆపరేటర్	అర్థం
1	: , (స్పేస్)	రిఫరెన్స్ ఆపరేటర్లు
2	–	రుణ గుర్తు (negation) (–5)
3	%	శాతం
4	^	ఘాతం
5	* /	గుణకారం, భాగహారం
6	+ –	కూడిక, తీసివేత
7	&	టెక్స్ట్‌ను కలపడం
8	= < > <= >= <>	పోల్చడం

బ్రాకెట్లు ఈ క్రమాన్ని మారుస్తాయి. ఒకే స్థాయిలో ఉంటే: ఎడమ నుంచి కుడికి.

ఉదాహరణ 2. =2+3*4^2/8* విలువ కనుక్కోండి. సాధన.* 4^2 = 16; 3 × 16 = 48; 48 ÷ 8 = 6; 2 + 6. **సమాధానం: 8**

ఉదాహరణ 3. =(5+3)*2^2* విలువ కనుక్కోండి. సాధన.* 8 × 4. **సమాధానం: 32**

3. ఫార్ములాలను కాపీ చేయడం: రిలేటివ్, అబ్సల్యూట్.

ఉదాహరణ 4. C1 లో =A1+B1 ఉంది. దాన్ని C2 కు, D1 కు కాపీ చేశారు. వాటిలో ఏముంటుంది? *సాధన.* ఒక అడ్డు వరుస కిందికి: =A2+B2. ఒక నిలువు వరుస కుడికి: =B1+C1. **సమాధానం: C2 =A2+B2; D1 =B1+C1**

ఉదాహరణ 5. C2 లో =$A2*B$1 ఉంది. దాన్ని *D3* కు కాపీ చేశారు. ఫలితం? *(a) =$A3*C$1 (b) =$A2*C$1 (c) =$B3*C$2 (d) =$A3*B$1* సాధన.* కదలిక: ఒక నిలువు వరుస కుడికి, ఒక అడ్డు వరుస కిందికి. $A2: నిలువు వరుస స్థిరం, అడ్డు వరుస 2 → 3. B$1: నిలువు వరుస B → C, అడ్డు వరుస స్థిరం. **సమాధానం: (a)**

ఉపయోగం: =B2*E1 అనేది B నిలువు వరుసలోని ప్రతి ధరను E1 లో ఉన్న ఒకే రేటుతో గుణిస్తుంది; కాపీ చేసినా అది E1 నే సూచిస్తుంది.

4. సాధారణ ఫంక్షన్లు. డేటా: A1 = 10, A2 = 20, A3 ఖాళీ, A4 = "abc", A5 = 30, A6 = 0.

ఫంక్షన్	చేసే పని	A1:A6 కు ఫలితం
SUM(A1:A6)	సంఖ్యలను కూడుతుంది	60
COUNT(A1:A6)	**సంఖ్యలను** లెక్కిస్తుంది	4

ఫంక్షన్	చేసే పని	A1:A6 కు ఫలితం
COUNTA(A1:A6)	**ఖాళీ కాని** సెల్‌లను లెక్కిస్తుంది	5
COUNTBLANK(A1:A6)	ఖాళీ సెల్‌లను లెక్కిస్తుంది	1
AVERAGE(A1:A6)	SUM ÷ COUNT	15
MAX / MIN	అతిపెద్దది / అతిచిన్నది	30 / 0

AVERAGE **ఖాళీ సెల్‌లను, టెక్స్ట్‌ను వదిలేస్తుంది** కానీ **సున్నాలను లెక్కలోకి తీసుకుంటుంది**: 60 ÷ 4 = 15, 60 ÷ 6 కాదు.

ఉదాహరణ 6. B2:B6 లో మార్కులు: 35, 62, 48, 90, 27 (ఉత్తీర్ణత మార్కు 40).

ఫార్ములా	ఫలితం
=AVERAGE(B2:B6)	52.4
=MEDIAN(B2:B6)	48
=COUNTIF(B2:B6,">=40")	3
=SUMIF(B2:B6,">=40")	200
=IF(B2>=40,"Pass","Fail")	Fail
=RANK(B4,B2:B6)	3

5. ప్రింటింగ్.

- **ప్రింట్ ప్రివ్యూ (Print Preview)** ముద్రణకు ముందే పేజీలను చూపుతుంది; **పేజ్ సెటప్ (Page Setup)** **ఓరియంటేషన్** (పోర్ట్రెయిట్/ల్యాండ్‌స్కేప్), కాగితం పరిమాణం, **మార్జిన్లు**, **స్కేలింగ్** ("Fit to 1 page wide") నిర్ణయిస్తుంది.
- **ప్రింట్ ఏరియా (Print Area)** ఎంపిక చేసిన రేంజ్‌ను మాత్రమే ముద్రిస్తుంది; **ప్రింట్ టైటిల్స్ (Print Titles)** ("Rows to repeat at top") శీర్షిక అడ్డు వరుసలను ప్రతి పేజీలోనూ మళ్లీ ముద్రిస్తుంది.
- **హెడర్, ఫుటర్**లలో పేజీ సంఖ్యలు, తేదీ, ఫైల్ పేరు ఉంటాయి.
- **గ్రిడ్‌లైన్లు, అడ్డు/నిలువు వరుసల శీర్షికలు డిఫాల్ట్‌గా ముద్రితం కావు**; పేజ్ సెటప్ → షీట్‌లో వాటిని ఆన్ చేయాలి.
- **పేజ్ బ్రేక్ ప్రివ్యూ (Page Break Preview)** పేజీ విభజనలను చూపుతుంది, జరుపుతుంది. షార్ట్‌కట్: **Ctrl+P**.

లోతైన విశ్లేషణ

1. గణాంక పనిలో వాడే మరిన్ని ఫంక్షన్లు.

ఫంక్షన్	ఉపయోగం	ఉదాహరణ → ఫలితం
ROUND(x, n)	n దశాంశ స్థానాలకు సవరించడం	ROUND(3.14159, 2) → 3.14
ROUND(x, −2)	వందలకు సవరించడం	ROUND(1234.5, −2) → 1200
INT(x)	పూర్ణాంకానికి **కిందికి** సవరించడం	INT(−3.7) → −4
MOD(a, b)	శేషం, b గుర్తుతో	MOD(17, 5) → 2; MOD(−7, 3) → 2

ఇతరాలు: **MEDIAN, MODE, STDEV** (నమూనా) / **STDEV.P** (సమష్టి), **VAR, CORREL, PERCENTILE, QUARTILE, GEOMEAN, HARMEAN, FREQUENCY** (పౌనఃపున్య పట్టికల కోసం అరే ఫంక్షన్), **SUMPRODUCT** (Σfx ఒకే దశలో), **VLOOKUP / HLOOKUP** (పట్టికలోని మొదటి నిలువు వరుస / అడ్డు వరుసలో ఒక విలువను వెతకడం), **AND, OR, NOT**, **CONCATENATE** లేదా &, **TODAY(), NOW()**.

ఉదాహరణ 7. A2:A6 లోని విలువలకు B2:B6 లో పౌనఃపున్యాలు ఉన్నాయి. సగటును ఒకే ఫార్ములాలో రాయండి. సాధన. సగటు = Σfx ÷ Σf. **సమాధానం: =SUMPRODUCT(A2:A6,B2:B6)/SUM(B2:B6)**

2. ఎర్రర్ విలువలు.

ఎర్రర్	కారణం
#DIV/0!	సున్నాతో (లేదా ఖాళీ సెల్‌తో) భాగించడం
#VALUE!	తప్పు రకం, ఉదా. అంకగణితంలో టెక్స్ట్
#REF!	తొలగించిన సెల్‌కు రిఫరెన్స్
#NAME?	ఫంక్షన్ లేదా పేరు తప్పుగా రాయడం
#N/A	విలువ అందుబాటులో లేదు (లుకప్ విఫలమైంది)
#NUM!	చెల్లని సంఖ్య (ఉదా. −1 వర్గమూలం)
#NULL!	ఒకదానినొకటి ఖండించని రెండు రేంజ్‌లు

తన సొంత సెల్‌నే (నేరుగా లేదా పరోక్షంగా) సూచించే ఫార్ములాను **సర్క్యులర్ రిఫరెన్స్ (circular reference)** అంటారు.

3. గందరగోళపరిచే ప్రాధాన్య క్రమం. ఎక్సెల్‌లో **రుణ గుర్తు ఘాతం కంటే ముందు** లెక్కిస్తారు: =-2^2 విలువ **4**, కానీ =0-2^2 విలువ **−4** (ఇక్కడ − తీసివేత). ఎక్సెల్ సగాన్ని **సున్నా నుంచి దూరంగా** సవరిస్తుంది: ROUND(2.5, 0) = 3, ROUND(−2.5, 0) = −3.

4. సెల్ రిఫరెన్స్ ఇతర రూపాలు.

- మొత్తం నిలువు వరుస **A:A**, మొత్తం అడ్డు వరుస **5:5**.
- మరో షీట్: **Sheet2!B4**; మరో వర్క్‌బుక్: **[Book2.xlsx]Sheet1!A1**.
- **యూనియన్ (union)** (కామా): SUM(A1:A3, C1:C3); **ఖండన (intersection)** (స్పేస్): SUM(A1:C3 B2:D4) ఉమ్మడి సెల్‌లు B2:C3 ను మాత్రమే కూడుతుంది.
- **పేరున్న రేంజ్ (named range):** B2:B6 కు "Marks" అని పేరు పెట్టి =AVERAGE(Marks) అని రాయవచ్చు.

5. డేటా సాధనాలు. సార్ట్ (Sort) (ఆరోహణ/అవరోహణ, బహుళ స్థాయి), **ఫిల్టర్ / ఆటోఫిల్టర్ (Filter / AutoFilter)** (సరిపోలే అడ్డు వరుసలను మాత్రమే చూపుతుంది; మిగిలినవి దాచబడతాయి, తొలగించబడవు), **డేటా వాలిడేషన్ (Data Validation)** (నమోదులను పరిమితం చేయడం, ఉదా. 0–100), **కండిషనల్ ఫార్మాటింగ్ (Conditional Formatting)** (నియమం ప్రకారం సెల్‌లకు రంగు), **పివట్ టేబుల్ (Pivot Table)** (క్రాస్-ట్యాబులేషన్, స్ప్రెడ్‌షీట్‌లో పట్టికీకరణ సాధనం), **గోల్ సీక్ (Goal Seek)** (లక్ష్య ఫలితాన్నిచ్చే ఇన్‌పుట్‌ను కనుక్కోవడం), **చార్ట్‌లు** (కాలమ్, బార్, లైన్, పై, స్కాటర్/XY, హిస్టోగ్రామ్).

తరచూ చేసే పొరపాట్లు

- ✗ "ఫార్ములాలు + తో మొదలవుతాయి." ✓ ఎక్సెల్, కాల్క్‌లలో అవి = తో మొదలవుతాయి.
- ✗ "COUNT ఖాళీ కాని అన్ని సెల్‌లను లెక్కిస్తుంది." ✓ COUNT **సంఖ్యలను** లెక్కిస్తుంది; **COUNTA** ఖాళీ కాని సెల్‌లను లెక్కిస్తుంది.
- ✗ "AVERAGE ఖాళీ సెల్‌లను సున్నాగా తీసుకుంటుంది." ✓ ఖాళీలను (టెక్స్ట్‌ను కూడా) **వదిలేస్తుంది**; టైప్ చేసిన 0 ను మాత్రం **లెక్కిస్తుంది**.
- ✗ "$A1 అడ్డు వరుసను స్థిరం చేస్తుంది." ✓ **$A1 నిలువు వరుసను స్థిరం చేస్తుంది**; **A$1** అడ్డు వరుసను స్థిరం చేస్తుంది.
- ✗ "#### అంటే ఫార్ములా దోషం." ✓ **నిలువు వరుస చాలా ఇరుకుగా ఉంది**.
- ✗ "గ్రిడ్‌లైన్లు డిఫాల్ట్‌గా ముద్రితమవుతాయి." ✓ **కావు**; పేజ్ సెటప్‌లో వాటిని ఆన్ చేయాలి.
- ✗ "వర్క్‌బుక్, వర్క్‌షీట్ ఒకటే." ✓ **వర్క్‌బుక్** (ఫైల్) లో **వర్క్‌షీట్లు** ఉంటాయి.
- ✗ "ఎక్సెల్ 2007+ లో 65,536 అడ్డు వరుసలు." ✓ **1,048,576 అడ్డు వరుసలు**, 16,384 నిలువు వరుసలు (XFD వరకు); 65,536 అనేది ఎక్సెల్ 2003.

1.4 కోడ్‌లు (BCD, ASCII, EBCDIC, Unicode), బిట్‌లు, బైట్‌లు, వర్డ్‌లు, కంప్యూటర్ మెమరీ

కంప్యూటర్లు ప్రతిదాన్నీ బిట్‌ల (bits) రూపంలో నిల్వ చేస్తాయి; ఏ బిట్ అమరిక ఏ అంకెను లేదా అక్షరాన్ని సూచిస్తుందో కోడ్‌లు (codes) (బీసీడీ (BCD), ఆస్కీ (ASCII), ఎబ్సిడిక్ (EBCDIC), యూనికోడ్ (Unicode)) నిర్ణయిస్తాయి; మెమరీ (memory) ఈ అమరికలను ఒక సోపాన క్రమంలో ఉంచుతుంది: వేగవంతమైన, చిన్న రిజిస్టర్లు, క్యాష్ నుంచి పెద్ద, నెమ్మదైన ద్వితీయ నిల్వ వరకు.

ముఖ్య వాస్తవాలు

- **బిట్** = బైనరీ అంకె (0 లేదా 1), అతిచిన్న యూనిట్. **నిబుల్ (nibble)** = 4 బిట్‌లు. **బైట్ (byte)** = 8 బిట్‌లు (ఆస్కీలో ఒక అక్షరం). **వర్డ్ (word)** = సీపీయూ (CPU) ఒకే యూనిట్‌గా నిర్వహించే బిట్‌ల సంఖ్య (16, 32 లేదా 64 బిట్‌లు; యంత్రాన్ని బట్టి మారుతుంది).
- **యూనిట్లు (పాఠ్యపుస్తకాలు పాటించే బైనరీ పద్ధతి):**

యూనిట్	సమానం	2 యొక్క ఘాతం
1 KB	1,024 బైట్‌లు	2^{10}
1 MB	1,024 KB	2^{20}
1 GB	1,024 MB	2^{30}
1 TB	1,024 GB	2^{40}

- TB తర్వాత: PB (2^{50}), EB (2^{60}), ZB (2^{70}), YB (2^{80}).
- **బీసీడీ (8421 కోడ్):** ప్రతి దశాంశ అంకెను **4 బిట్‌లలో** రాస్తారు; 1010 నుంచి 1111 వాడరు (చెల్లవు).
- **ఎక్సెస్-3 (excess-3):** ప్రతి అంకెకు బీసీడీ + 3 (0011); ఇది **స్వీయ పూరక (self-complementing)** కోడ్.
- **గ్రే కోడ్ (Gray code):** వరుస విలువల మధ్య **ఒక్క బిట్ మాత్రమే** మారుతుంది (యూనిట్-దూర కోడ్, రిఫ్లెక్టెడ్ బైనరీ కోడ్).
- **ఆస్కీ:** అమెరికన్ స్టాండర్డ్ కోడ్ ఫర్ ఇన్ఫర్మేషన్ ఇంటర్‌చేంజ్ (American Standard Code for Information Interchange), **7-బిట్**, **128** అక్షరాలు; విస్తరించిన ఆస్కీ 8 బిట్‌లు వాడుతుంది (256).
- **ఎబ్సిడిక్:** ఎక్స్‌టెండెడ్ బైనరీ కోడెడ్ డెసిమల్ ఇంటర్‌చేంజ్ కోడ్ (Extended Binary Coded Decimal Interchange Code), **8-బిట్**, మెయిన్‌ఫ్రేమ్‌ల కోసం **ఐబీఎం (IBM)** రూపొందించింది.
- **ఇస్కీ (ISCII):** ఇండియన్ స్క్రిప్ట్ కోడ్ ఫర్ ఇన్ఫర్మేషన్ ఇంటర్‌చేంజ్ (Indian Script Code for Information Interchange), 8-బిట్, **1991** నాటి బీఐఎస్ (BIS) ప్రమాణం.
- **యూనికోడ్:** ప్రపంచంలోని ప్రతి లిపికీ ఒకే కోడ్; కోడ్ పాయింట్‌లు (code points) **U+0000 నుంచి U+10FFFF** (1,114,112). ఎన్‌కోడింగ్‌లు: **యూటీఎఫ్-8 (UTF-8)** (1 నుంచి 4 బైట్‌లు, ఆస్కీతో అనుకూలం),

యూటీఎఫ్-16 (UTF-16) (2 లేదా 4 బైట్‌లు), **యూటీఎఫ్-32 (UTF-32)** (4 బైట్‌లు).

- **ఆస్కీ విలువలు:** 'A' = **65**, 'a' = **97**, '0' = **48**, ఖాళీ (space) = 32.
- **ప్రాథమిక మెమరీ (primary memory):** ర్యాండమ్ యాక్సెస్ మెమరీ (RAM) (వోలటైల్ (volatile), చదవడం/రాయడం), రీడ్ ఓన్లీ మెమరీ (ROM) (నాన్-వోలటైల్ (non-volatile), చదవడం మాత్రమే). **ద్వితీయ మెమరీ (secondary memory):** హార్డ్ డిస్క్, ఎస్ఎస్డీ (SSD), పెన్ డ్రైవ్, సీడీ/డీవీడీ (CD/DVD), టేప్ (నాన్-వోలటైల్).
- **అత్యంత వేగం నుంచి అత్యంత నెమ్మది:** రిజిస్టర్లు → క్యాష్ → ర్యామ్ → ద్వితీయ నిల్వ.

1. సంఖ్యల కోడ్‌లు.

ఉదాహరణ 1. 59 ను బీసీడీలో, శుద్ధ బైనరీలో రాయండి. సాధన. బీసీడీ ప్రతి అంకెకు విడిగా కోడ్ ఇస్తుంది: 5 = 0101, 9 = 1001 → **0101 1001** (8 బిట్‌లు). శుద్ధ బైనరీ: 59 = **111011** (6 బిట్‌లు). బీసీడీకి ఎక్కువ బిట్‌లు కావాలి కానీ దశాంశ ప్రదర్శనకు, దాని నుంచి సులభంగా మారుతుంది.

ఉదాహరణ 2. దశాంశ 5 యొక్క ఎక్సెస్-3 కోడ్, బైనరీ 1011 యొక్క గ్రే కోడ్ కనుగొనండి. సాధన. 5 + 3 = 8 → **1000**. గ్రే: మొదటి బిట్‌ను (1) అలాగే ఉంచి, తర్వాత ప్రతి పక్క పక్క జతకు XOR చేయండి: 1⊕0 = 1, 0⊕1 = 1, 1⊕1 = 0 → **1110**.

2. అక్షరాల కోడ్‌లు. పెద్ద అక్షరాలు 'A' నుంచి 'Z' 65 నుంచి 90; చిన్న అక్షరాలు 'a' నుంచి 'z' 97 నుంచి 122; చిన్న అక్షరం = దాని పెద్ద అక్షరం **+ 32**. అంకెలు '0' నుంచి '9' 48 నుంచి 57, కాబట్టి '7' అనే అక్షరం 7 అనే సంఖ్య కాదు.

ఉదాహరణ 3. 'D' యొక్క ఆస్కీ కోడ్ దశాంశంలో, బైనరీలో (a) 68, 1000100 (b) 100, 1100100 (c) 67, 1000011 (d) 68, 1000010 సాధన. 'A' = 65, కాబట్టి 'D' = 68 = 64 + 4. **సమాధానం: (a)**

కోడ్‌ల పోలిక

కోడ్	బిట్‌లు	అక్షరాలు	ఉపయోగం
ఆస్కీ	7	128	పీసీలు, ఇంటర్నెట్ పాఠ్యం
ఎబ్సిడిక్	8	256	ఐబీఎం మెయిన్‌ఫ్రేమ్‌లు
ఇస్కీ	8	భారతీయ లిపులు	పాత భారతీయ సాఫ్ట్‌వేర్
యూనికోడ్	8/16/32 యూనిట్లు	అన్ని లిపులు	ఆధునిక వ్యవస్థలు, వెబ్

తెలుగుకు యూనికోడ్‌లో సొంత బ్లాక్ **U+0C00 నుంచి U+0C7F** ఉంది; అందుకే ఫోన్‌లో టైప్ చేసిన తెలుగు పాఠ్యం ఏ కంప్యూటర్‌లోనైనా సరిగ్గా కనిపిస్తుంది.

3. పారిటీ బిట్ (parity bit). 1ల సంఖ్య సరి (**సరి పారిటీ (even parity)**) లేదా బేసి (**బేసి పారిటీ (odd parity)**) అయ్యేలా చేర్చే అదనపు బిట్; ఇది ఒక్క బిట్ దోషాన్ని గుర్తిస్తుంది కానీ సరిచేయలేదు.

ఉదాహరణ 4. 1011001 కు సరి-పారిటీ బిట్ కనుగొనండి. సాధన. దీనిలో నాలుగు 1లు ఉన్నాయి, ఇప్పటికే సరి. **సమాధానం: 0** (బేసి పారిటీకి 1 కావాలి)

4. మెమరీ రకాలు.

రకం	వోలటైలా?	లక్షణం
ఎస్‌ర్యామ్ (SRAM)	అవును	ఫ్లిప్-ఫ్లాప్‌లు, వేగవంతం, రిఫ్రెష్ అక్కర్లేదు; క్యాష్‌లో వాడతారు
డీర్యామ్ (DRAM)	అవును	కెపాసిటర్లు, రిఫ్రెష్ చేయాలి; ప్రధాన మెమరీ
రోమ్ / పీరోమ్ (PROM)	కాదు	పీరోమ్‌ను వినియోగదారు ఒక్కసారే రాస్తారు
ఈపీరోమ్ (EPROM) / ఈఈపీరోమ్ (EEPROM)	కాదు	అతినీలలోహిత కాంతితో / విద్యుత్‌తో చెరిపేస్తారు

- **ఫ్లాష్ మెమరీ (flash memory)** బ్లాక్‌లుగా చెరిపే ఒక రకమైన ఈఈపీరోమ్; పెన్ డ్రైవ్‌లు, మెమరీ కార్డులు, ఎస్ఎస్‌డీలలో వాడతారు.
- **ఫర్మ్‌వేర్ (firmware)** (బయోస్/యూఈఎఫ్ఐ (BIOS/UEFI) ప్రారంభ ప్రోగ్రామ్) రోమ్‌లో లేదా ఫ్లాష్‌లో ఉంటుంది.
- **క్యాష్ మెమరీ (cache memory)** సీపీయూ, ర్యామ్ మధ్య ఉంటుంది; స్థాయిలు L1 (అత్యంత వేగం, అతిచిన్నది), L2, L3.
- **రిజిస్టర్లు (registers)** సీపీయూ లోపల ఉంటాయి (అక్యుమ్యులేటర్, ప్రోగ్రామ్ కౌంటర్, ఇన్‌స్ట్రక్షన్ రిజిస్టర్).
- **ద్వితీయ నిల్వ:** హార్డ్ డిస్క్ (అయస్కాంత; ట్రాక్‌లు, సెక్టార్‌లు, సిలిండర్‌లు), ఎస్ఎస్‌డీ (ఫ్లాష్, కదిలే భాగాలు లేవు), ఆప్టికల్ డిస్క్‌లు (సీడీ సుమారు 700 MB, డీవీడీ 4.7 GB సింగిల్ లేయర్, బ్లూ-రే 25 GB సింగిల్ లేయర్), అయస్కాంత టేప్ (**సీక్వెన్షియల్ యాక్సెస్ (sequential access)**, బ్యాకప్‌లు).

లోతైన విశ్లేషణ

- **మెమరీ సోపాన క్రమం (memory hierarchy):** కిందికి వెళ్ళేకొద్దీ (రిజిస్టర్ → క్యాష్ → ర్యామ్ → డిస్క్ → టేప్) **వేగం తగ్గుతుంది, సామర్థ్యం పెరుగుతుంది, ఒక్కో బిట్ ఖర్చు తగ్గుతుంది**. ప్రోగ్రామ్‌లు దగ్గరలోని డేటాను మళ్ళీ మళ్ళీ వాడతాయి కాబట్టి (సమీప వినియోగ సూత్రం (locality of reference)) ఈ క్రమం పనిచేస్తుంది; క్యాష్‌లోనే దొరికే అభ్యర్థనల భాగమే **హిట్ నిష్పత్తి (hit ratio)**.
- **యాక్సెస్:** ర్యామ్ **ర్యాండమ్ యాక్సెస్ (random access)** (ఏ స్థానానికైనా ఒకే సమయం); టేప్ **సీక్వెన్షియల్**; డిస్క్ **డైరెక్ట్ (సెమీ-ర్యాండమ్) యాక్సెస్ (direct access)**. డిస్క్ యాక్సెస్ సమయం = **సీక్ టైమ్ (seek time) + రొటేషనల్ లేటెన్సీ (rotational latency)** (+ బదిలీ సమయం).
- **వర్చువల్ మెమరీ (virtual memory):** హార్డ్ డిస్క్‌లో కొంత భాగాన్ని ర్యామ్‌కు పొడిగింపుగా వాడటం; దీంతో ర్యామ్ కంటే పెద్ద ప్రోగ్రామ్‌లు కూడా నడుస్తాయి; పేజీలు లోపలికి, బయటికి మార్చబడతాయి.
- **అడ్రస్ లైన్లు, సామర్థ్యం:** n అడ్రస్ లైన్లు (address lines) **2^n** స్థానాలను ఎంచుకుంటాయి; సామర్థ్యం = 2^n × వర్డ్ పరిమాణం.
- **ఆచరణలో యూనిట్లు:** డిస్క్ తయారీదారులు దశాంశ యూనిట్లు వాడతారు (1 GB = 10^9 బైట్‌లు); ఐఈసీ (IEC) 2^{10}, 2^{20}, 2^{30} లకు KiB, MiB, GiB అనే పేర్లు ఇచ్చింది. వేరే చెప్పకపోతే పరీక్షలు 1 KB = 1,024 బైట్‌లు అనే

పాఠ్యపుస్తక పద్ధతినే పాటిస్తాయి.

ఉదాహరణ 5. ఒక మెమరీ చిప్‌కు 16 అడ్రస్ లైన్లు, 8-బిట్ వర్డ్ ఉన్నాయి. దాని సామర్థ్యం ఎంత? (a) 16 KB (b) 32 KB (c) 64 KB (d) 128 KB సాధన. 2^{16} = 65,536 స్థానాలు × 1 బైట్ = 65,536 బైట్లు = 64 KB. **సమాధానం: (c) 64 KB**

ఉదాహరణ 6. 32-బిట్ బైట్ అడ్రస్ ఉన్న సీపీయూ ఎంత మెమరీని అడ్రస్ చేయగలదు? సాధన. 2^{32} బైట్లు = $2^{2} \times 2^{30}$ బైట్లు. **సమాధానం: 4 GB**

ఉదాహరణ 7. 2 KB లో ఎన్ని బిట్లు ఉంటాయి? సాధన. 2 × 1,024 × 8. **సమాధానం: 16,384 బిట్లు**

ఉదాహరణ 8. వీటిలో నాన్-వోలటైల్ ఏది? (a) ఎస్‌ర్యామ్ (b) డీర్యామ్ (c) క్యాష్ (d) ఈఈపీరోమ్ సాధన. కరెంటు పోతే ర్యామ్, క్యాష్ డేటాను కోల్పోతాయి. **సమాధానం: (d) ఈఈపీరోమ్**

తరచూ చేసే పొరపాట్లు

- ✗ "ఆస్కీ 256 అక్షరాల 8-బిట్ కోడ్." ✓ ప్రామాణిక ఆస్కీ **7-బిట్, 128**; 8-బిట్ అనేది విస్తరించిన ఆస్కీ.
- ✗ "ఎబ్సిడిక్‌ను ఏఎన్ఎస్ఐ (ANSI) రూపొందించింది." ✓ **ఐబీఎం**; ఆస్కీ అమెరికన్ ప్రమాణాల సంస్థ నుంచి వచ్చింది.
- ✗ "59 యొక్క బీసీడీ = 111011." ✓ అది శుద్ధ బైనరీ; బీసీడీ **0101 1001**.
- ✗ "1 KB = 1,000 బైట్లు" (కంప్యూటర్ బేసిక్స్ పేపర్‌లో). ✓ **1,024 బైట్లు**.
- ✗ "వర్డ్ ఎప్పుడూ 16 బిట్లు." ✓ వర్డ్ పొడవు **ప్రాసెసర్‌ను బట్టి మారుతుంది**.
- ✗ "క్యాష్‌కు డీర్యామ్ వాడతారు." ✓ క్యాష్ **ఎస్‌ర్యామ్** వాడుతుంది; డీర్యామ్‌ను రిఫ్రెష్ చేయాలి, అది ప్రధాన మెమరీ.
- ✗ "ఈపీరోమ్‌ను విద్యుత్‌తో చెరిపేస్తారు." ✓ ఈపీరోమ్‌ను **అతినీలలోహిత కాంతితో** చెరిపేస్తారు; ఈఈపీరోమ్‌ను విద్యుత్‌తో.
- ✗ "'a' = 65." ✓ **'A' = 65, 'a' = 97**.

1.5 బైనరీ అంకగణితం, పూరకాలు; లాజిక్ గేట్లు, తార్కిక ప్రక్రియలు

ఏఎల్‌యూ (ALU) బైనరీ సంఖ్యలపై అంకగణిత ప్రక్రియలు (కూడిక, 2 యొక్క పూరకాన్ని (2's complement) కలపడం ద్వారా తీసివేత), బిట్ వారీగా తార్కిక ప్రక్రియలు (AND, OR, NOT, XOR) చేస్తుంది; ఇవి లాజిక్ గేట్లతో (logic gates) నిర్మితమవుతాయి; NAND లేదా NOR ఒక్కటే మిగతా అన్ని గేట్లనూ నిర్మించగలదు.

ముఖ్య వాస్తవాలు

- **బైనరీ కూడిక:** 0 + 0 = 0; 0 + 1 = 1; 1 + 1 = **10** (0, క్యారీ (carry) 1); 1 + 1 + 1 = **11** (1, క్యారీ 1).
- **బైనరీ తీసివేత:** 0 − 0 = 0; 1 − 0 = 1; 1 − 1 = 0; 0 − 1 = **1, బారో (borrow) 1**.

- **గుణకారం:** 0 × 0 = 0 × 1 = 0, 1 × 1 = 1 (జరిపి కలపడం). **భాగహారం:** పదే పదే జరిపి తీసివేయడం.
- **1 యొక్క పూరకం (1's complement):** ప్రతి బిట్‌ను (bit) తిప్పండి (0 ↔ 1). **2 యొక్క పూరకం** = 1 యొక్క పూరకం **+ 1**.
- **దశాంశ సమానాలు:** 9 యొక్క పూరకం (ప్రతి అంకెను 9 నుంచి తీసివేయడం), 10 యొక్క పూరకం (9 యొక్క పూరకం + 1).
- **కంప్యూటర్లు కలపడం ద్వారా తీసివేస్తాయి:** తీసివేయవలసిన సంఖ్య యొక్క 2 యొక్క పూరకాన్ని కలుపుతాయి.
- **n-బిట్ 2 యొక్క పూరకం పరిధి (range): -2^{n-1} నుంచి $2^{n-1} - 1$**; 8 బిట్‌లు: **−128 నుంచి +127**. ఎంఎస్‌బీ (MSB) సంజ్ఞ బిట్ (sign bit) (0 = +, 1 = −).
- **ప్రాథమిక గేట్‌లు:** ఆండ్ (AND), ఆర్ (OR), నాట్ (NOT). **సార్వత్రిక గేట్‌లు (universal gates): నాండ్ గేట్ (NAND), నార్ గేట్ (NOR)**. ఇతరాలు: ఎక్స్‌ఓఆర్ (XOR), ఎక్స్‌ఎన్‌ఓఆర్ (XNOR).

గేట్	సమీకరణం	అవుట్‌పుట్ 1 ఎప్పుడు
AND	A·B	అన్ని ఇన్‌పుట్‌లు 1 అయినప్పుడు
OR	A + B	కనీసం ఒక ఇన్‌పుట్ 1 అయినప్పుడు
NOT	A′	ఇన్‌పుట్ 0 అయినప్పుడు
NAND	(A·B)′	అన్ని ఇన్‌పుట్‌లు 1 కానప్పుడు
NOR	(A + B)′	అన్ని ఇన్‌పుట్‌లు 0 అయినప్పుడు
XOR	A ⊕ B	ఇన్‌పుట్‌లు వేరుగా ఉన్నప్పుడు
XNOR	(A ⊕ B)′	ఇన్‌పుట్‌లు సమానంగా ఉన్నప్పుడు

- n ఇన్‌పుట్‌లకు (inputs) **సత్య పట్టికలో (truth table) 2^n** అడ్డు వరుసలు ఉంటాయి.
- **డీ మోర్గన్ (De Morgan):** (A + B)′ = A′·B′, (A·B)′ = A′ + B′.

ఉదాహరణ 1. 10110_2, 01101_2 లను కలపండి. సాధన. 22 + 13 = 35. క్యారీలతో నిలువు వరుస వారీగా కలిపితే 100011 వస్తుంది. **సమాధానం: 100011_2**

1. నాలుగు ప్రక్రియలు.

ఉదాహరణ 2. $10110_2 - 01101_2$ కనుగొనండి. సాధన. 22 − 13 = 9. **సమాధానం: 1001_2**

ఉదాహరణ 3. 1011_2 ను 101_2 తో గుణించండి. సాధన. 1011 × 1 = 1011; తర్వాతి బిట్ 0 → 00000; తర్వాతి బిట్ 1 → 101100. మొత్తం 1011 + 101100 = 110111. సరిచూడటం: 11 × 5 = 55. **సమాధానం: 110111_2**

ఉదాహరణ 4. 110110_2 ను 110_2 తో భాగించండి. సాధన. 54 ÷ 6 = 9. **సమాధానం: 1001_2**

2. పూరకాలు.

ఉదాహరణ 5. 01011010 యొక్క 1 యొక్క పూరకం, 2 యొక్క పూరకం కనుగొనండి. సాధన. తిప్పితే: 10100101. 1 కలిపితే: 10100110. **సమాధానం: 10100101, 10100110** 2 యొక్క పూరకానికి సులభ మార్గం: కుడి నుంచి మొదటి 1 వరకు (దానితో సహా) బిట్‌లను అలాగే రాసి, మిగతావాటిని తిప్పండి.

ఉదాహరణ 6. −25 ను 8-బిట్ 2 యొక్క పూరకంలో రాయండి. (a) 10011001 (b) 11100110 (c) 11100111 (d) 00011001 సాధన. +25 = 00011001; తిప్పితే → 11100110; + 1 → 11100111. **సమాధానం: (c)** ((a) సంజ్ఞ-పరిమాణం (sign-magnitude), (b) 1 యొక్క పూరకం)

3. పూరకాల ద్వారా తీసివేత (4 బిట్‌లు).

ఉదాహరణ 7. 2 యొక్క పూరకంతో 9 − 5. సాధన. 5 = 0101 → 2 యొక్క పూరకం 1011. 1001 + 1011 = 1 0100. బయటికి క్యారీ వస్తే ఫలితం ధనాత్మకం: **దానిని వదిలేయండి**. **సమాధానం: 0100 = +4**

ఉదాహరణ 8. 2 యొక్క పూరకంతో 5 − 9. సాధన. 9 = 1001 → 2 యొక్క పూరకం 0111. 0101 + 0111 = 1100, **క్యారీ లేదు**, కాబట్టి సమాధానం రుణాత్మకం: 1100 యొక్క 2 యొక్క పూరకం తీసుకోండి = 0100. **సమాధానం: −4**

ఉదాహరణ 9. 1 యొక్క పూరకంతో 9 − 5. సాధన. 0101 యొక్క 1 యొక్క పూరకం = 1010. 1001 + 1010 = 1 0011. క్యారీని తిరిగి కలపండి (**ఎండ్-అరౌండ్ క్యారీ (end-around carry)**): 0011 + 1 = 0100. **సమాధానం: +4**

ఉదాహరణ 10. 10 యొక్క పూరకంతో 8754 − 4567. సాధన. 4567 యొక్క 10 యొక్క పూరకం = 5433. 8754 + 5433 = 14187; క్యారీ 1 ని వదిలేయండి. **సమాధానం: 4187**

4. తార్కిక (బిట్-వారీ) ప్రక్రియలు. ఏఎల్‌యూ ప్రతి బిట్‌ల జతకు ఒక గేట్‌ను వర్తింపజేస్తుంది.

A = 1100, B = 1010 అయితే:

ప్రక్రియ	ఫలితం
A AND B	1000
A OR B	1110
A XOR B	0110
NOT A	0011

ఉపయోగాలు: AND ఎంచుకున్న బిట్‌లను **సున్నా చేస్తుంది** (మాస్కింగ్ (masking)), OR బిట్‌లను **1 చేస్తుంది**, XOR బిట్‌లను **తిప్పుతుంది**; A XOR A = 0. స్ప్రెడ్‌షీట్‌లు (spreadsheets) ఇదే తర్కాన్ని AND(), OR(), NOT() ఫంక్షన్లలో (functions) వాడతాయి; అవి TRUE లేదా FALSE ఇస్తాయి.

5. సార్వత్రిక గేట్‌లు. ప్రతి గేట్‌ను నాండ్ గేట్‌లతో మాత్రమే (లేదా నార్ గేట్‌లతో మాత్రమే) నిర్మించవచ్చు:

కావలసిన గేట్	NAND గేట్‌లు	NOR గేట్‌లు
NOT	1	1
AND	2	3
OR	3	2
XOR	4	5

లోతైన విశ్లేషణ

- **8 బిట్‌లలో −5 యొక్క సంజ్ఞా రూపాలు:** సంజ్ఞ-పరిమాణం **10000101**, 1 యొక్క పూరకం **11111010**, 2 యొక్క పూరకం **11111011**. సంజ్ఞ-పరిమాణం, 1 యొక్క పూరకంలో **రెండు సున్నాలు** (+0, −0) ఉంటాయి, పరిధి −127 నుంచి +127; 2 యొక్క పూరకంలో **ఒకే సున్నా**, పరిధి −128 నుంచి +127; అందుకే కంప్యూటర్లు దీనినే వాడతాయి.
- **ఓవర్‌ఫ్లో (overflow): ఒకే సంజ్ఞ** గల రెండు సంఖ్యలను కలిపితే వ్యతిరేక సంజ్ఞ గల ఫలితం వచ్చినప్పుడు జరుగుతుంది. 8 బిట్‌లలో 100 + 50 = 150 > 127 అయితే 10010110 వస్తుంది, ఇది **−106** గా చదవబడుతుంది. వ్యతిరేక సంజ్ఞల సంఖ్యలను కలిపితే ఎప్పుడూ ఓవర్‌ఫ్లో జరగదు.
- **బూలియన్ బీజగణిత (Boolean algebra) నియమాలు:**
  - తత్సమ (identity) A + 0 = A, A·1 = A; శూన్య (null) A + 1 = 1, A·0 = 0.
  - ఇడెంపొటెంట్ (idempotent) A + A = A, A·A = A; పూరక (complement) A + A′ = 1, A·A′ = 0; ద్వి-నిషేధ (involution) (A′)′ = A.
  - శోషణ (absorption) A + AB = A, A(A + B) = A; అలాగే A + A′B = A + B.
  - విభాగ (distributive) A + BC = (A + B)(A + C) బూలియన్ బీజగణితంలో నిజం (సాధారణ బీజగణితంలో కాదు).
  - **ద్వంద్వత (duality):** + ను · తో, 0 ను 1 తో మార్చితే మరో చెల్లుబాటయ్యే నియమం వస్తుంది.
- **యాడర్‌లు (adders): హాఫ్ యాడర్ (half adder)** 2 బిట్‌లను కలుపుతుంది: మొత్తం = A ⊕ B, క్యారీ = A·B. **ఫుల్ యాడర్ (full adder)** 3 బిట్‌లను (క్యారీ ఇన్ C తో) కలుపుతుంది: మొత్తం = A ⊕ B ⊕ C, క్యారీ = AB + BC + CA. హాఫ్ సబ్‌ట్రాక్టర్ (half subtractor) తేడా A ⊕ B, బారో A′B ఇస్తుంది.
- **బహుళ ఇన్‌పుట్ XOR** 1 ఇన్‌పుట్‌ల సంఖ్య **బేసి** అయినప్పుడు 1 ఇస్తుంది (పారిటీ బిట్ ఇలాగే తయారవుతుంది).
- **జరపడం (shifts):** ఎడమకు ఒక స్థానం జరిపితే 2 తో గుణించినట్లు; కుడికి జరిపితే 2 తో భాగించినట్లు (సంజ్ఞ లేని పూర్ణ సంఖ్యలకు).
- n చరరాశుల (variables) **బూలియన్ ప్రమేయాల సంఖ్య** = $2^{(2^n)}$; 2 చరరాశులకు 16.

ఉదాహరణ 11. AB + AB′ ను సూక్ష్మీకరించండి. సాధన. A(B + B′) = A·1. **సమాధానం: A**

ఉదాహరణ 12. రెండు ఇన్‌పుట్‌లూ 0 అయినప్పుడు మాత్రమే అవుట్‌పుట్ 1 ఇచ్చే గేట్ ఏది? (a) NAND (b) NOR (c) XOR (d) XNOR సాధన. NOR 0, 0 కు మాత్రమే 1 ఇస్తుంది. XNOR 1, 1 కు కూడా 1 ఇస్తుంది. **సమాధానం: (b) NOR**

ఉదాహరణ 13. $3A_{16} + 2F_{16}$, $57_8 + 26_8$ లను కలపండి. సాధన. హెక్స్: A + F = 10 + 15 = 25 = 16 + 9 → 9 రాసి, క్యారీ 1; 3 + 2 + 1 = 6 → 69_{16} (58 + 47 = 105). ఆక్టల్: 7 + 6 = 13 = 8 + 5 → 5 రాసి, క్యారీ 1; 5 + 2 + 1 = 8 = 10_8 → 105_8 (47 + 22 = 69). **సమాధానం: 69_{16}, 105_8**

తరచూ చేసే పొరపాట్లు

- ✗ "బైనరీలో 1 + 1 = 2." ✓ 1 + 1 = **10** (0 రాసి, క్యారీ 1).
- ✗ "2 యొక్క పూరకం = బిట్‌లను తిప్పడం." ✓ అది **1 యొక్క** పూరకం; 2 యొక్క పూరకంలో **1** కలుపుతారు.
- ✗ "2 యొక్క పూరకంతో తీసివేతలో చివరి క్యారీని తిరిగి కలపాలి." ✓ దానిని **వదిలేయాలి**; ఎండ్-అరౌండ్ క్యారీ **1 యొక్క** పూరకానికి.
- ✗ "క్యారీ లేకపోతే సమాధానం ధనాత్మకం." ✓ క్యారీ లేకపోతే ఫలితం **రుణాత్మకం**, పూరక రూపంలో ఉంటుంది.
- ✗ "8-బిట్ సంజ్ఞా పరిధి −127 నుంచి +128." ✓ **−128 నుంచి +127**.
- ✗ "AND, OR, NOT సార్వత్రిక గేట్‌లు." ✓ సార్వత్రిక గేట్‌లు **NAND, NOR**.
- ✗ "రెండు ఇన్‌పుట్‌లూ 1 అయినప్పుడు XOR 1 ఇస్తుంది." ✓ ఇన్‌పుట్‌లు **వేరుగా** ఉన్నప్పుడు XOR 1; 1, 1 కు అది 0.
- ✗ "(A + B)′ = A′ + B′." ✓ డీ మోర్గన్: **(A + B)′ = A′·B′**.

త్వరిత పునశ్చరణ

పరీక్షకు ముందు రోజు ఇది చదవండి

1.1 అల్గారిథం, ఫ్లోచార్ట్: గుర్తులు, లక్షణాలు, ఉదాహరణలు

- అల్గారిథం: పరిమితం, నిర్దిష్టం, ప్రభావశీలం; 0+ ఇన్‌పుట్‌లు, 1+ అవుట్‌పుట్‌లు.
- అండాకారం = ప్రారంభం/ముగింపు; సమాంతర చతుర్భుజం = ఇన్‌పుట్/అవుట్‌పుట్; దీర్ఘచతురస్రం = ప్రక్రియ.
- వజ్రాకారం = నిర్ణయం (రెండు నిష్క్రమణలు); వృత్తం = అదే పేజీ కనెక్టర్; పంచభుజి = ఆఫ్-పేజ్ కనెక్టర్.
- ప్రవాహం పై నుంచి కిందికి, ఎడమ నుంచి కుడికి; బాణాలు దిశను చూపుతాయి.
- మూడు నిర్మాణాలు: క్రమం, ఎంపిక, పునరావృతం.
- while: ముందు పరీక్ష (0+ సార్లు); do-while: చివర పరీక్ష (1+ సార్లు).
- డ్రై రన్ = ట్రేస్ పట్టికతో చేతితో అనుసరించడం.
- బైనరీ శోధనకు క్రమబద్ధీకరించిన డేటా కావాలి; సుమారు $\log_2 n$ పోలికలు.
- బబుల్ సార్ట్: $n - 1$ విడతలు, $n(n-1)/2$ పోలికలు.
- యూక్లిడ్: గ.సా.భా.(48, 18) = 6.

1.2 సంఖ్యా విధానాలు: బైనరీ, ఆక్టల్, హెక్సాడెసిమల్, దశాంశంలోకి, దశాంశం నుంచి మార్పిడి

- బైనరీ 2, ఆక్టల్ 8, దశాంశం 10, హెక్స్ 16; అంకెలు 0 నుంచి ఆధారం − 1.
- A నుంచి F = 10 నుంచి 15.
- దశాంశంలోకి: Σ అంకె × స్థాన విలువ.
- పూర్ణ భాగం: ÷ ఆధారం, శేషాలు కింది నుంచి పైకి.
- భిన్న భాగం: × ఆధారం, పూర్ణాంక భాగాలు పై నుంచి కిందికి.
- 3 బిట్‌లు = 1 ఆక్టల్ అంకె; 4 బిట్‌లు = 1 హెక్స్ అంకె.
- n బిట్‌లు → 2^n అమరికలు, గరిష్ఠం $2^n - 1$.
- $45 = 101101_2 = 55_8 = 2D_{16}$.
- $255 = 11111111_2 = 377_8 = FF_{16}$.
- $0.625 = 0.101_2$; 0.1 బైనరీలో పునరావృతం అవుతుంది.

1.3 స్ప్రెడ్‌షీట్లు: సెల్‌లు, రేంజ్‌లు, ఫార్ములాలు, సెల్ రిఫరెన్స్‌లు, ఫంక్షన్లు, ఎడిటింగ్, ప్రింటింగ్

- వర్క్‌బుక్ = ఫైల్; వర్క్‌షీట్ = గ్రిడ్; సెల్ = నిలువు వరుస అక్షరం + అడ్డు వరుస సంఖ్య.
- రేంజ్ A1:C4 = 12 సెల్‌లు; నేమ్ బాక్స్ యాక్టివ్ సెల్‌ను చూపుతుంది.
- ఫార్ములాలు = తో మొదలవుతాయి; SUM, AVERAGE, COUNT, IF వంటి ఫంక్షన్లు.
- రిలేటివ్ A1; అబ్సల్యూట్ A1; మిక్స్‌డ్ $A1 / A$1; F4 తో మార్చవచ్చు.
- ప్రాధాన్య క్రమం: రుణ గుర్తు, %, ^, * /, + –, &, పోలికలు.
- COUNT = సంఖ్యలు; COUNTA = ఖాళీ కానివి; AVERAGE ఖాళీలను వదిలేస్తుంది, సున్నాలను లెక్కిస్తుంది.
- #DIV/0!, #VALUE!, #REF!, #NAME?, #N/A; #### = ఇరుకైన నిలువు వరుస.
- ఎక్సెల్ 2007+: 1,048,576 అడ్డు వరుసలు × 16,384 నిలువు వరుసలు (XFD).
- ప్రింటింగ్: ప్రింట్ ఏరియా, ప్రింట్ టైటిల్స్, పేజ్ సెటప్, ప్రింట్ ప్రివ్యూ, Ctrl+P.
- గ్రిడ్‌లైన్లు డిఫాల్ట్‌గా ముద్రితం కావు.

1.4 కోడ్‌లు (BCD, ASCII, EBCDIC, Unicode), బిట్‌లు, బైట్‌లు, వర్డ్‌లు, కంప్యూటర్ మెమరీ

- నిబుల్ 4 బిట్‌లు, బైట్ 8 బిట్‌లు, వర్డ్ = సీపీయూ యూనిట్ (యంత్రాన్ని బట్టి).
- KB 2^{10}, MB 2^{20}, GB 2^{30}, TB 2^{40} బైట్‌లు.
- బీసీడీ: ప్రతి దశాంశ అంకెకు 4 బిట్‌లు; ఎక్సెస్-3 = బీసీడీ + 3.
- గ్రే కోడ్: ఒకసారికి ఒక్క బిట్ మాత్రమే మారుతుంది.
- ఆస్కీ 7-బిట్ (128); ఎబ్సిడిక్ 8-బిట్ (ఐబీఎం); ఇస్కీ 1991; యూనికోడ్ అన్ని లిపులు.
- 'A' 65, 'a' 97, '0' 48.
- పారిటీ బిట్ ఒక్క బిట్ దోషాన్ని గుర్తిస్తుంది.
- ఎస్‌ర్యామ్ క్యాష్, డీర్యామ్ ప్రధాన మెమరీ; ఈపీరోమ్ అతినీలలోహిత కాంతి, ఈఈపీరోమ్ విద్యుత్.
- రిజిస్టర్లు → క్యాష్ → ర్యామ్ → డిస్క్: నెమ్మది, పెద్దది, చౌక.
- n అడ్రస్ లైన్లు → 2^n స్థానాలు.

1.5 బైనరీ అంకగణితం, పూరకాలు; లాజిక్ గేట్‌లు, తార్కిక ప్రక్రియలు

- 1 + 1 = 10; 1 + 1 + 1 = 11; 0 − 1 = 1 బారో 1.
- 1 యొక్క పూరకం: తిప్పడం; 2 యొక్క పూరకం: తిప్పి + 1.
- 2 యొక్క పూరకంతో తీసివేత: క్యారీ వస్తే → వదిలేయండి, ధనాత్మకం; క్యారీ లేకపోతే → రుణాత్మకం, మళ్ళీ పూరకం తీసుకోండి.
- 1 యొక్క పూరకం: ఎండ్-అరౌండ్ క్యారీ.
- 8-బిట్ సంజ్ఞా సంఖ్యలు: −128 నుంచి +127; −25 = 11100111.
- ఓవర్‌ఫ్లో: ఒకే సంజ్ఞ ఇన్‌పుట్‌లు, వ్యతిరేక సంజ్ఞ ఫలితం.
- సార్వత్రిక గేట్‌లు NAND, NOR; XOR = 4 NAND లు.
- డీ మోర్గన్: (A + B)′ = A′B′; (AB)′ = A′ + B′.
- హాఫ్ యాడర్: మొత్తం A ⊕ B, క్యారీ AB.
- సత్య పట్టిక: 2^n అడ్డు వరుసలు.

ప్రాక్టీస్ ప్రశ్నలు

18 ప్రశ్నలు · జవాబులు చివరలో

1. Consider the following statements:

1. Each octal digit can be written as a group of exactly three binary digits.

2. Each hexadecimal digit can be written as a group of exactly four binary digits.

Which of the statements given above is/are correct?

a. 1 only

b. 2 only

c. Both 1 and 2

d. Neither 1 nor 2

2. Consider the following statements about Unicode:

1. The first 128 Unicode code points are the same as the ASCII characters.

2. UTF-8 always uses exactly 2 bytes for every character.

Which of the statements given above is/are correct?

a. 1 only

b. 2 only

c. Both 1 and 2

d. Neither 1 nor 2

3. Consider the following statements about the XNOR gate:

1. Its output is 1 when both inputs are equal.

2. It is the complement of the XOR gate.

3. For inputs A and B its output is A·B + A'·B'.

Which of the statements given above are correct?

a. 1 and 2 only

b. 2 and 3 only

c. 1 and 3 only

d. 1, 2 and 3

4. Match List I (chart type) with List II (best use):

List I:

A. Pie chart

B. Line chart

C. Column chart

D. Scatter (XY) chart

List II:

1. Trend of a value over time

2. Relationship between two numeric variables

3. Share of each part in a whole

4. Comparison of values across categories

Select the correct answer using the code given below.

a. A-3, B-4, C-1, D-2

b. A-3, B-1, C-4, D-2

c. A-4, B-1, C-3, D-2

d. A-3, B-1, C-2, D-4

5. What is the octal equivalent of the hexadecimal number $(3A7)_{16}$?

a. 1637

b. 935

c. 1747

d. 1647

6. Consider the following statements:

1. The 4-bit groups 1010 to 1111 are invalid in BCD (8421) code.

2. BCD usually needs more bits than pure binary to represent the same number.

3. Gray code is a weighted code.

Which of the statements given above are correct?

a. 1 and 3 only

b. 1 and 2 only

c. 2 and 3 only

d. 1, 2 and 3

7. Which of the following pairs of flowchart symbol and its use is NOT correctly matched?

a. Rectangle with double vertical side lines : predefined process (a sub-routine defined elsewhere)

b. Arrow : direction of flow

c. Small circle : start or stop of the flowchart

d. Rectangle with a wavy bottom edge : printed document or report

8. Columns in a worksheet are labelled A, B, ..., Z, AA, AB, ... What is the column number of column AB?

a. 27 b. 28

c. 29 d. 54

9. Consider the following statements:

1. In a four-digit hexadecimal integer, the place value of the leftmost position is 16^3.
2. The hexadecimal letters A to F stand for the decimal values 10 to 15.
3. 8 is a valid digit in the octal system.
4. The binary system uses only two symbols, 0 and 1.

Which of the statements given above are correct?

a. 1 and 2 only b. 2, 3 and 4 only

c. 1, 2 and 4 only d. 1, 2, 3 and 4

10. The Gray code 1101 is converted to binary. What is the result?

a. 1011 b. 1001

c. 1101 d. 0110

11. Consider the following statements:

1. In a rotate operation no bit is lost; the bit moved out at one end comes back at the other end.
2. Shifting an unsigned number left by one place doubles it, provided no 1 bit is moved out of the leftmost position.

Which of the statements given above is/are correct?

a. 1 only b. 2 only

c. Both 1 and 2 d. Neither 1 nor 2

12. In MS-Excel, cell C2 contains the formula =$A2*B$1. This cell is copied and pasted into cell E5. Which formula will E5 contain?

a. =$A2*D$4 b. =$C5*D$1

c. =$A5*B$1 d. =$A5*D$1

13. Match List I (number) with List II (decimal value):

List I:

A. $(1110)_2$

B. $(21)_8$

C. $(1C)_{16}$

D. $(101011)_2$

List II:

1. 17

2. 43

3. 14

4. 28

Select the correct answer using the code given below.

a. A-3, B-4, C-1, D-2

b. A-1, B-3, C-4, D-2

c. A-3, B-1, C-2, D-4

d. A-3, B-1, C-4, D-2

14. Consider the following statements:

1. BCD (8421) is a weighted code.

2. Excess-3 is a weighted code.

Which of the statements given above is/are correct?

a. 1 only

b. 2 only

c. Both 1 and 2

d. Neither 1 nor 2

15. Using 8-bit 2's complement arithmetic, 00111100 (60) is subtracted from 00101101 (45) by adding the 2's complement of 60. What is the 8-bit result?

a. 00001111

b. 11110000

c. 10001111

d. 11110001

16. Match List I (number system) with List II (base):

List I:

A. Binary

B. Octal

C. Decimal

D. Hexadecimal

List II:

1. 16

2. 2

3. 10

4. 8

Select the correct answer using the code given below.

a. A-2, B-4, C-3, D-1

b. A-2, B-1, C-3, D-4

c. A-4, B-2, C-3, D-1

d. A-2, B-4, C-1, D-3

17. Using the binary convention (1 KB = 1024 bytes), which of the following pairs is NOT correctly matched?

a. 1 nibble : 4 bits

b. 1 byte : 8 bits

c. 1 MB : 1024 bytes

d. 1 GB : 1024 MB

18. A memory chip is organised as 32K × 8 bits (32K locations of one byte each). How many address lines does one chip need, and how many such chips are needed to build a 256 KB byte-addressable memory?

a. 8 address lines; 8 chips

b. 15 address lines; 8 chips

c. 15 address lines; 64 chips

d. 32 address lines; 8 chips

ಜವಾಬುಲು: `1-(c) 2-(a) 3-(d) 4-(b) 5-(d) 6-(b) 7-(c) 8-(b) 9-(c) 10-(b) 11-(c) 12-(d) 13-(d) 14-(a) 15-(d) 16-(a) 17-(c) 18-(b)`