
**Senior Division
Cells**

PROBLEM: The ACSL cell always has 8-character bits. The bits are always some combination of A, B, C, D, E, F, G, and H. The cell performs operations as listed below:

DIVIDE - The cell divides into two cells with one cell taking the first 4 bits and arranging them alphabetically and the second cell taking the last 4 bits and arranging them alphabetically. Then each partial cell replicates and concatenates to get back to 8 bits each.

e.g. **DIVIDE** AHBGCEDF becomes ABGHABGH and CDEFCDEF

ADD_n - The first n ($0 \leq n \leq 4$) bits replicate, are put in alphabetical order and then concatenated to the first n bits on the right. The last n bits are deleted. Remaining bits are concatenated on the right.

e.g. **ADD₃** AHBGCEDF becomes AHBABHGC

SUBTRACT_n - The first n ($0 \leq n \leq 4$) bits are deleted and the last n bits replicate, are put in alphabetical order and then are concatenated on the right.

e.g. **SUBTRACT₃** AHBGCEDF becomes GCEDFDEF

UNION - Two cells become one by deleting the first four bits of the first cell and the last four bits of the second cell. The remaining bit groups are put in alphabetical order and the remaining bits of the second cell are concatenated to the right of the remaining bits of the first cell.

e.g. **UNION** AHBGCEDF, AGBHCFED becomes CDEFABGH

INTERSECT - Two cells become one by deleting the middle four bits of the first cell and the middle four bits of the second cell. The remaining bit groups are put in alphabetical order and the remaining bits of the second cell are concatenated to the right of the remaining bits of the first cell.

e.g. **INTERSECT** AHBGCEDF, AGBHCFED becomes ADFHADEG

INPUT: There will be 5 lines of input. Each line will contain an operation followed by a string(s) representing the 8 bit cell(s).

OUTPUT: Print the outcome of the operation on the cell.

SAMPLE INPUT

1. **DIVIDE**, AHBGCEDF
2. **ADD₃**, AHBGCEDF
3. **SUBTRACT₃**, AHBGCEDF
4. **UNION**, AHBGCEDF, AGBHCFED
5. **INTERSECT**, AHBGCEDF, AGBHCFED

SAMPLE OUTPUT

1. ABGHABGH and CDEFCDEF
2. AHBABHGC
3. GCEDFDEF
4. CDEFABGH
5. ADFHADEG