#include <stdio.h>

#include <math.h>

// Function to calculate parity bits

void calculateParityBits(int data[], int hamming[], int m, int r) {

int i, j, k, parity;

for (i = 0; i < r; i++) {

parity = 0;

k = pow(2, i);

for (j = k - 1; j < m + r; j += k \* 2) {

for (int x = 0; x < k && (j + x) < m + r; x++) {

parity ^= hamming[j + x];

}

}

hamming[k - 1] = parity;

}

}

int main() {

int m, r = 0, i, j;

// Get the number of data bits

printf("Enter the number of data bits: ");

scanf("%d", &m);

int data[m];

printf("Enter %d data bits (one at a time):\n", m);

for (i = 0; i < m; i++) {

scanf("%d", &data[i]);

}

// Determine the number of parity bits needed

while (pow(2, r) < (m + r + 1)) {

r++;

}

int hamming[m + r];

// Place data bits in the correct positions

j = 0;

for (i = 0; i < m + r; i++) {

if ((i + 1) && !(i & (i + 1))) {

hamming[i] = 0; // Placeholder for parity bit

} else {

hamming[i] = data[j++];

}

}

// Calculate parity bits

calculateParityBits(data, hamming, m, r);

// Display the Hamming Code

printf("Hamming code: ");

for (i = 0; i < m + r; i++) {

printf("%d ", hamming[i]);

}

printf("\n");

return 0;

}