#include <bits/stdc++.h>

using namespace std;

struct Tree{

int left, right, parent;

int bukva;

};

Tree tree [512];

int main()

{

FILE \*f;

FILE \*g;

int nuli;

int size\_tree;

f = fopen("output.haff", "rb");

g = fopen("razoutput.haff", "wb");

freopen("razoutput.haff", "w", stdout);

fscanf(f, "%d", &nuli);

fscanf(f, "%d", &size\_tree);

for (int i = 0; i < size\_tree; i++) {

fscanf(f, "%d", &tree[i].left);

fscanf(f, "%d", &tree[i].right);

fscanf(f, "%d", &tree[i].parent);

fscanf(f, "%d", &tree[i].bukva);

}

size\_tree--;

unsigned char ch;

fscanf(f, "\*");

int pos = size\_tree;

unsigned char maska = 128;

while (fscanf(f, "%c", &ch) != -1) {

maska = 128;

for (int i = 0; i < 8; i++) {

if ((ch & maska) == maska) {

pos = tree[pos].right;

if (tree[pos].right == -1 && tree[pos].left == -1) {

char x = tree[pos].bukva;

cout << x;

pos = size\_tree;

}

} else {

pos = tree[pos].left;

if (tree[pos].right == -1 && tree[pos].left == -1) {

char x = tree[pos].bukva;

cout << x;

pos = size\_tree;

}

}

maska = maska >> 1;

}

}

return 0;

}