Alexs-iMac:9 alex$ cat main.c

#include<stdio.h>

int absR(int a)

{

return a > 0 ? a : -a;

}

int modR(int a, int b)

{

return (a % b + absR(b)) % absR(b);

}

int divR(int a, int b)

{

return (a - modR(a, b)) / b;

}

int signR(int a)

{

return a > 0 ? 1 : (a < 0 ? -1 : 0);

}

int maxR(int a, int b)

{

return a > b ? a : b;

}

int minR(int a, int b)

{

return a < b ? a : b;

}

int i\_P(int i, int j, int l, int k)

{

return minR(modR(l, 5), modR(i \* k, 5)) + j + divR(k, 3);

}

int j\_P(int i, int j, int l, int k)

{

return divR(maxR(-3 \* i, 2 \* j), 5) - absR(j - l);

}

int l\_P(int i, int j, int l, int k)

{

return j + modR(l, 7) + modR(k \* signR(i), 10);

}

int main(void)

{

const int sideS = 10;

const int I\_S = 10;

const int J\_S = -10;

int i\_p;

int j\_p;

int l\_p;

int i;

int j;

int l;

int k = 0;

scanf("%d%d%d", &i, &j, &l);

if (I\_S - sideS / 2 <= i && i <= I\_S + sideS / 2 && J\_S - sideS / 2 <= j && j <= J\_S + sideS / 2) {

printf("Yes\n%d %d %d %d\n", i, j, l, k);

return 0;

}

while (k < 50) {

i\_p = i\_P(i, j, l, k);

j\_p = j\_P(i, j, l, k);

l\_p = l\_P(i, j, l, k);

i = i\_p;

j = j\_p;

l = l\_p;

k++;

if (I\_S - sideS / 2 <= i && i <= I\_S + sideS / 2 && J\_S - sideS / 2 <= j && j <= J\_S + sideS / 2) {

printf("Yes\n%d %d %d %d\n", i, j, l, k);

return 0;

}

}

printf("No\n%d %d %d %d\n", i, j, l, k);

return 0;

}

Alexs-iMac:9 alex$ gcc -std=c99 -pedantic -Wall main.c

Alexs-iMac:9 alex$ ./a.out

10 10 -10

Yes

13 -5 18 5

Alexs-iMac:9 alex$