```
#include <iostream>
#include <algorithm>
using namespace std;

int v[100];

void toArr(long long n, int& length) {
      length = 0;
      while (n > 0) {
            v[length] = n % 10;
            n /= 10;
            length++;
      }
      reverse(v, v + length);
}

int firstDigit(int n) {
      int result = 0;
      while (n > 0) {
            result = n % 10;
            n /= 10;
      }
      return result;
}

struct Test {
      long long n;
      int p;
      int d;

      void read() {
            cin >> n >> p >> d;
      }

      void solve() {
            int length = 0;

            toArr(n, length);

            int c = v[length - p];
            if (c >= 0 && c <= 4) {
                  v[length - p] = (v[length - p] + d) % 10;
            }
            else if (c >= 5 && c <= 9) {
                  v[length - p] = firstDigit(abs(v[length - p] - d));
            }

            for (int i = length - p + 1; i < length; i++) {
                  v[i] = 0;
            }

            for (int i = 0; i < length; i++) {
                  cout << v[i];
            }
```

```
        cout << '\n';
    }
};


int main() {

    Test* arr = new Test[5];

    for (int i = 0; i < 5; i++) {
        arr[i].read();
    }

    for (int i = 0; i < 5; i++) {
        arr[i].solve();
    }

    return 0;
}
```