```
#include<stdio.h>
#include<string.h>

#define MAX 112345
#define left(p) (p) << 1
#define right(p) ((p) << 1) + 1

typedef struct { int hom, ele, rat; }jogo;

int n, lazy[4 * MAX];
jogo st[4 * MAX];

void build(int p, int l, int r) {
  int meio = (l + r) / 2;
  if (l == r) { st[p].hom = 1; st[p].ele = st[p].rat = 0; return; }
  build(left(p), l, meio);
  build(right(p), meio + 1, r);
  st[p].hom = st[left(p)].hom + st[right(p)].hom;
  st[p].ele = st[left(p)].ele + st[right(p)].ele;
  st[p].rat = st[left(p)].rat + st[right(p)].rat;
}

void range_update(int p, int l, int r, int i, int j) {
  int aux, meio = (l + r) / 2, md;
  if (lazy[p]) {
    md = lazy[p] % 3;
    if (md == 1) {
      aux = st[p].ele; st[p].ele = st[p].hom;
      st[p].hom = st[p].rat; st[p].rat = aux;
    } else if (md == 2) {
      aux = st[p].rat; st[p].rat = st[p].hom;
      st[p].hom = st[p].ele; st[p].ele = aux;
    }
    if (l != r) {
      lazy[left(p)] += lazy[p];
      lazy[right(p)] += lazy[p];
    }
    lazy[p] = 0;
  }
  if (i > r || j < l) return;
  if (i <= l && j >= r) {
    aux = st[p].ele; st[p].ele = st[p].hom;
    st[p].hom = st[p].rat; st[p].rat = aux;
    if (l != r) {
      lazy[left(p)] += 1;
      lazy[right(p)] += 1;
    }
    return;
  }
  range_update(left(p), l, meio, i, j);
  range_update(right(p), meio + 1, r, i, j);
  st[p].hom = st[left(p)].hom + st[right(p)].hom;
  st[p].ele = st[left(p)].ele + st[right(p)].ele;
  st[p].rat = st[left(p)].rat + st[right(p)].rat;
}

jogo rmq(int p, int l, int r, int i, int j) {
  int meio = (l + r) / 2, aux, md;
  jogo p1, p2, ret;
  if (i > r || j < l) { ret.hom = -1; return ret; }
  if (lazy[p]) {
    md = lazy[p] % 3;
    if (md == 1) {
      aux = st[p].ele; st[p].ele = st[p].hom;
      st[p].hom = st[p].rat; st[p].rat = aux;
    } else if (md == 2) {
      aux = st[p].rat; st[p].rat = st[p].hom;
      st[p].hom = st[p].ele; st[p].ele = aux;
    }
    if (l != r) {
      lazy[left(p)] += lazy[p];
      lazy[right(p)] += lazy[p];
    }
    lazy[p] = 0;
  }
  if (l >= i && r <= j) return st[p];
  p1 = rmq(left(p), l, meio, i, j);
  p2 = rmq(right(p), meio + 1, r, i, j);
  if (p1.hom == -1) return p2;
  if (p2.hom == -1) return p1;
  ret.hom = p1.hom + p2.hom;
  ret.ele = p1.ele + p2.ele;
  ret.rat = p1.rat + p2.rat;
  return ret;
}

int main(void) {
  int m, a, b;
  char c;
  jogo resp;
  while (scanf("%d %d", &n, &m) != EOF) {
    memset(lazy, 0, sizeof(lazy)); build(1, 0, n - 1);
    while (m--) {
      scanf(" %c %d %d", &c, &a, &b); a--; b--;
      if (c == 'M') range_update(1, 0, n - 1, a, b);
      else {
        resp = rmq(1, 0, n - 1, a, b);
        printf("%d %d %d\n", resp.hom, resp.ele, resp.rat);
      }}
    printf("\n");
  }
  return 0;
}
```