整数座標の点 (xs, ys) から点 (xe, ye) までの線分を描画するプログラム片は，線分の描画は xd >= yd の (x 成分の差が y 成分の差よりも大きい) 場合のみを考えると，以下のとおりである．

```
int xd = xe - xs;
int yd = ye - ys;
if (xd >= yd) {
    double d = ((double)yd)/xd;
    for (int x = xs; x <= xe; x++) {
        int y = (int)(d*x + 0.5);
        drawpoint(x, y);
    }
}
```

ここで drawpoint(x, y) は，点 (x, y) のピクセルを描画する命令である．このプログラムをもとに，同様の線分を描画する DDA(Digital Differential Analyzer) のプログラム (ブレゼンハムのアルゴリズム) を導く過程を考える．

(1) y 成分の変化を整数部分 y と小数部分 e (より正確には (小数部分 - 1)) に分けて計算すると，次のようになる．

```
int xd = xe - xs;
int yd = ye - ys;
if (xd >= yd) {
    double d = ((double)yd)/xd;
    double e = -0.5;
    for (int x = xs, y = ys; x <= xe; x++) {
        drawpoint(x, y);
        e += d;
        if (e > 0.0) {
            y++;  e -= 1.0;
        }
    }
}
```

(2) (1) のプログラム片から浮動小数点数を取り除くために，d，s の値を xd*2 倍する．すると次のプログラムが得られる．

```
int xd = xe - xs;
int yd = ye - ys;
if (xd >= yd) {
    int d = yd*2;
    int e = -xd;
    for (int x = xs, y = ys; x <= xe; x++) {
        drawpoint(x, y);
        e += d;
        if (e > 0) {
            y++;  e -= xd*2;
```

```
        }
    }
}
```

(3) (2) のプログラムにおける積演算をシフト演算に置き換えるとともに，そのシフト演算を 2 回だけに限定し，反復する命令数も減らすことを考える．その結果，プログラムは以下のようになる．

```
int xd = xe - xs;
int yd = ye - ys;
if (xd >= yd) {
    int d = yd << 1;
    int dr = d - (xd << 1);
    int e = d - xd;
    for (int x = xs, y = ys; x <= xe; x++) {
        drawpoint(x, y);
        if (e > 0) {
            y++;  e += dr;
        }
        else {
            e += d;
        }
    }
}
```