**public** **static** **void** water(**int** x1, **int** x2, **int** x3, **int** x4, **int** x5) {

/\*

\* (int)'y' = Sudoku(fbf) (int)'s' = Sudoku(ffd) (int)' ' = Sudoku(cb)

\* 121 111 117 114 32 108 117 99 107 121 32 107 101 121 32 104 97 118

\* 101 32 116 111 32 97 100 100 32 111 100 100 32 97 110 100 32 101 118

\* 101 110

\*

\* 116 97 107 101 32 121 111 117 114 32 107 101 121 32 97 110 100 32 121

\* 111 117 114 32 97 110 103 101 108 32 105 115 32 105 110 32 116 104

\* 101 32 110 111 114 116 104

\*/

**int** array[] = { x1, x2, x3, x4, x5 };

**int**[] buffer = **new** **int**[array.length];

**int** temp = 0;

**for** (**int** i = 0; i < 4; i++) {

**for** (**int** j = 0; j <= i; j++) {

**if** (array[j] > array[j + 1]) {

temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

}

m(array, buffer, 0, array.length - 1);

System.*out*.println("107 ,101 ,121 ,32" + ":" + *getM*(array, 5));

}

**public** **static** **int** getS(**int** a, **int** b) {

**return** (a \* b) / *getD*(a, b);

}

**public** **static** **int** getM(**int** num[], **int** n) {

**if** (n == 1)

**return** num[n - 1];

**return** *getS*(num[n - 1], *getM*(num, n - 1));

}

**public** **static** **int** getD(**int** a, **int** b) {

**if** (b == 0)

**return** a;

**return** *getD*(b, a % b);

}

**public** **static** **void** m(**int**[] array, **int**[] buffer, **int** start, **int** end) {

**int** length = end - start + 1;

**if** (length < 2) {

**return**;

}

**int** middle = length / 2 + start;

**int** ls = start;

**int** le = middle - 1;

**int** rs = middle;

**int** re = end;

m(array, buffer, ls, le);

m(array, buffer, rs, re);

**int** p = start;

**while** (ls <= le && rs <= re) {

buffer[p++] = array[ls] < array[rs] ? array[ls++] : array[rs++];

}

**while** (ls <= le) {

buffer[p++] = array[ls++];

}

**while** (rs <= re) {

buffer[p++] = array[rs++];

}

System.*arraycopy*(buffer, start, array, start, length);

}