```
a = {}

set a after adding 5 = {5}

set a after adding 5 = {5}

set a after adding 0 and 63 = {0, 5, 63}

set b = {1, 3, 6}

set b after removing 3 = {1, 6}

set b after removing 3 = {1, 6}

set b after removing 10 = {1, 6}

set b after removing 6 = {1}

set b after removing 1 = {}

set b after adding {10,2} = {2, 10}

a[0]={0, 1, 62}

a[1]={0, 1, 62}

a[2]={0, 1, 62}

a[3]={0, 1, 62}

a[4]={0, 1, 62}

TESTING: iset64 operator+(const iset64& a, const iset64& b)
Set a {1, 2}

Set b {1, 2, 3}

a + b = {1, 2, 3}

TESTING:iset64 operator+(const iset64& a, const int b)
{1, 2}

{1,2} + 1 = {1, 2}

{1, 2}

{1,2} + 3 = {1, 2, 3}

TESTING:iset64 operator+(const int b, const iset64& a)
Set a {1, 2}
```

```
 1 + {1,2} = {1, 2}

Set a {1, 2}

 3 + {1,2}  = {1, 2, 3}

TESTING:iset64& iset64::operator+=(const iset64& a)
Set b {1, 2}

Set a {1, 3}

 {1,2} + {1,3}  = {1, 2, 3}

iset64& iset64::operator+=(const int b)
Set a {1, 2}

 {1,2} + 3  = {1, 2, 3}

Set a {1, 2}

Set b {3, 4}

Set c {7, 8}

Set d {1, 2, 3, 4, 5, 7, 8}

TESTING: iset64 operator-(const iset64& a, const iset64& b)
Set a {1, 2}

Set b {1, 2}

a - b = {}

TESTING: iset64 operator-(const iset64& a, const iset64& b)
Set a {1, 5}

Set b {1, 2, 3}

a - b = {5}

TESTING: iset64 operator-(const iset64& a, const int b)
Set a {1, 2}

a - 3 = {1, 2}

TESTING: iset64 operator-(const int b, const iset64& a)
Set a {1, 2}

3 - a = {1, 2}
```

```
TESTING: iset64& iset64::operator-=(const iset64& a)
Set a {1, 3}

Set b {1, 2}

b -= a = {2}

TESTING: iset64& iset64::operator-=(const int b)
Set a {1, 2}

a -= 3 = {1, 2}

Set a {1, 2}

Set b {2, 4}

Set c {2, 8}

Set d {1, 5}

TESTING: iset64 operator*(const iset64& a, const iset64& b)
Set a {1, 2}

Set b {1, 2, 3}

a * b = {1, 2}

TESTING:iset64 operator*(const iset64& a, const int b)
Set a {1, 2}

{1,2} * 1 = {1}

Set a {1, 2}

{1,2} * 3 = {}

TESTING:iset64 operator*(const int b, const iset64& a)
Set a {1, 2}

 1 * {1,2} = {1}

Set a {1, 2}

 3 * {1,2}  = {}

TESTING:iset64& iset64::operator*=(const iset64& a)
Set b {1, 2}

Set a {1, 3}
```

```
 {1,2} * {1,3}  = {1}

iset64& iset64::operator*=(const int b)
Set a {1, 2}

 {1,2} * 3  = {}

Set a {1, 2}

Set b {2, 4}

Set c {2, 8}

Set d {2, 5}

TESTING: bool operator==(const iset64& a, const iset64& b)
Set a {1, 2}

Set b {1, 2}

a == b true
{1, 2}
{2}
a == b false
TESTING: bool operator!=(const iset64& a, const iset64& b)
Set a {1, 2}

Set b {1, 2}

a != b false
Set a {1, 2}

Set b {2}

a != b false
a = {1, 2, 63}

++a = {0, 2, 3}

a = {1, 2, 63}

acopy = {1, 2, 63}

a++ = {0, 2, 3}

rhs = {1, 2, 63}

a = {0, 2, 63}
```

```
--a = {1, 62, 63}

a = {0, 2, 63}

acopy = {0, 2, 63}

a-- = {1, 62, 63}

rhs = {0, 2, 63}

a = {0, 2, 63}

~a = {1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
54, 55, 56, 57, 58, 59, 60, 61, 62}

ans = {1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
54, 55, 56, 57, 58, 59, 60, 61, 62}

~ans = {0, 2, 63}

a = {0, 2, 63}

a exists
b = {}

b does not exists
a = {0, 2, 63}

a exists
b = {}

b does not exists
a = {4, 5, 6}

b = {5, 6, 8}

aplusb = {4, 5, 6, 8}

aplusbbar = {0, 1, 2, 3, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
54, 55, 56, 57, 58, 59, 60, 61, 62, 63}

abar = {0, 1, 2, 3, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
```

54, 55, 56, 57, 58, 59, 60, 61, 62, 63}

bbar = {0, 1, 2, 3, 4, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53,
54, 55, 56, 57, 58, 59, 60, 61, 62, 63}

abarplusbbar = {0, 1, 2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63}

abardotbbar = {0, 1, 2, 3, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35,
36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52,
53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63}

adotb = {5, 6}

adotbbar = {0, 1, 2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63}

Demorgan law (a+b)' = a'. b' is proved
Demorgan law (a.b)' = a' + b' is proved
a = {1, 2, 4, 5}

b = {2, 3, 5, 6}

aplusb = {1, 2, 3, 4, 5, 6}

aplusbbar = {0, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,
38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
55, 56, 57, 58, 59, 60, 61, 62, 63}

abar = {0, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,
38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
55, 56, 57, 58, 59, 60, 61, 62, 63}

bbar = {0, 1, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,
38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
55, 56, 57, 58, 59, 60, 61, 62, 63}

abarplusbbar = {0, 1, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,

```
51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63}

abardotbbar = {0, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37,
38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
55, 56, 57, 58, 59, 60, 61, 62, 63}

adotb  = {2, 5}

adotbbar  = {0, 1, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63}

Demorgan law (a+b)' = a'. b' is proved
Demorgan law (a.b)' = a' + b' is proved
Must attach output of the program to get a grade
Must attach a doc that explains the data structure that was used to
solve to get a grade
```