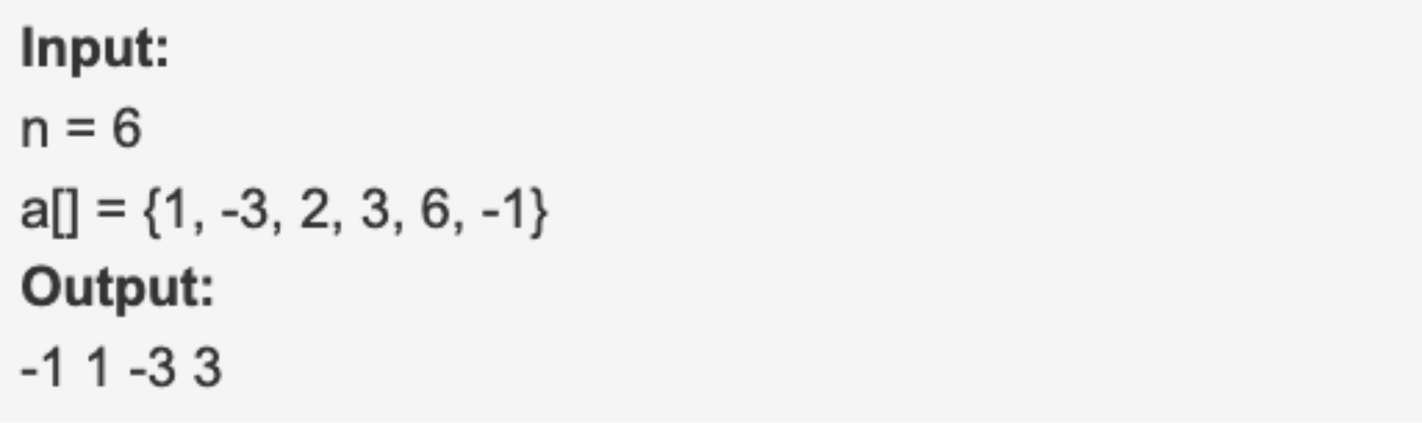
HASHING DATA STRUCTURE

- **Pairs with Positive Negative values**
- **pairs having positive value and negative value of a number**
- **If there is tie, print lexicographically smaller name.**



Input:

n = 6

a[] = {1, -3, 2, 3, 6, -1}

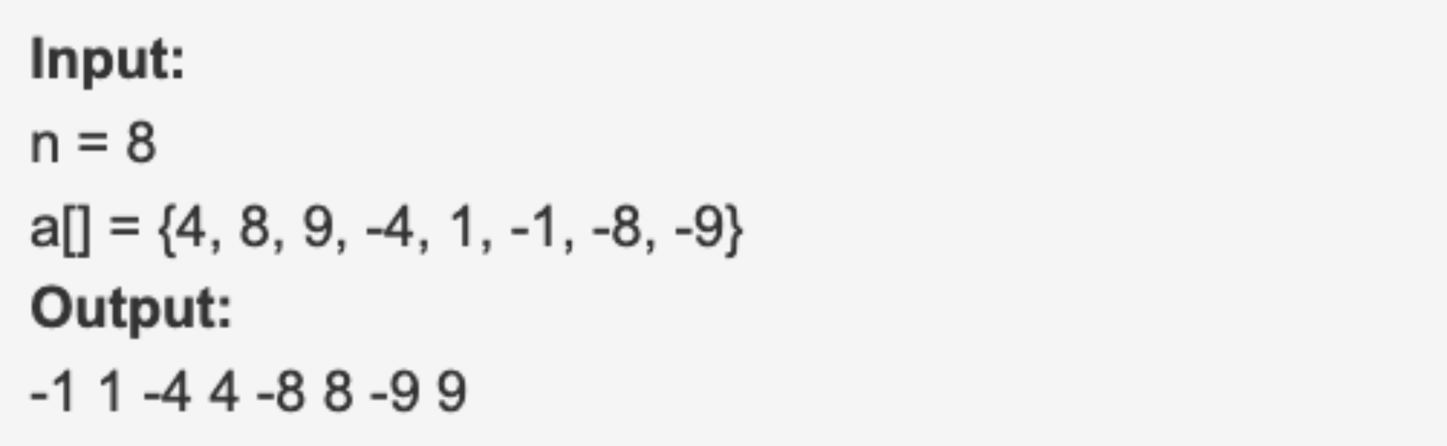
Output:

-1 1 -3 3

Hello World

HASHING DATA STRUCTURE

- **Pairs with Positive Negative values**
- **pairs having positive value and negative value of a number**
- **If there is tie, print lexicographically smaller name.**



Input:

n = 8

a[] = {4, 8, 9, -4, 1, -1, -8, -9}

Output:

-1 1 -4 4 -8 8 -9 9

Hello World

HASHING DATA STRUCTURE

- **1st solution**

- **unordered_set Positive**

  unordered_set Negative

- **Int n = 12**

  Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

  unordered_set <int> s

  NOTE: it contains unique value

HASHING DATA STRUCTURE

□ 1st solution

□ unordered_set Positive

unordered_set Negative

□ Int n = 12

Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

Positive = { 11, 4, 1, 2 }

unordered_set <int> s

NOTE: it contains unique value

o/p : -2, 2, -4, 4, -4, 4

Hello World

HASHING DATA STRUCTURE

□ 2nd solution

□ unordered_map Positive

unordered_map Negative

□ Int n = 12

Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

Start traversing positive map

Vector < int> ans

{ 4 4 4 4 }

Postive	Negative
11 -> 1	-4 -> 3
4 -> 2	-3 -> 1
2 -> 2	-9 -> 1
1 -> 1	-2 -> 1

Hello World

HASHING DATA STRUCTURE

- 2nd solution

- unordered_map Positive

  unordered_map Negative

- Int n = 12

  Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

Postive	Negative
11 -> 1	-4 -> 3
4 -> 2	-3 -> 1
2 -> 2	-9 -> 1
1 -> 1	-2 -> 1

Start traversing positive map

Vector < int> ans

{ 4 4 4 4 2 2 }

Hello world

HASHING DATA STRUCTURE

- 2nd solution

- unordered_map Positive

  unordered_map Negative

- Int n = 12

  Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

Postive	Negative
11 -> 1	-4 -> 3
4 -> 2	-3 -> 1
2 -> 2	-9 -> 1
1 -> 1	-2 -> 1

Start traversing positive map

Vector < int> ans

{ 4 4 4 4 2 2 }

Sort -> { 2 2 4 4 4 4 }

Multiply even index with -1

{ -2 2 -4 4 -4 4 }

Hello World

HASHING DATA STRUCTURE

- 3rd solution

- Int n = 12

  Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

- vector<int> v, ans ;

  Traverse array and store the negative value

  v = { -4, -3, -9, -4, -2, -4 }

map<int, int> mp;

1 -> 1

2 -> 2

4 -> 2

11 -> 1

Hello World

HASHING DATA STRUCTURE

- 3rd solution

- Int n = 12

  Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

- vector<int> v, ans ;

  Traverse array and store the negative value

  v = { -4, -3, -9, -4, -2, -4 }

  Sort the Array

  v = { -9, -4, -4, -4, -3, -2 }

map<int, int> mp;

1 -> 1

2 -> 2

4 -> 2

11 -> 1

Hello World

HASHING DATA STRUCTURE

- 3rd solution

- Int n = 12

  Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

- vector<int> v, ans ;

  Traverse array and store the negative value

  v = { -4, -3, -9, -4, -2, -4 }

  Sort the Array

  v = { -9, -4, -4, -4, -3, -2 }

  Traverse from behind

  o/p: { -2, 2, }

map<int, int> mp;

1 -> 1

2 -> 2

4 -> 2

11 -> 1

Hello World

HASHING DATA STRUCTURE

- 3rd solution

- Int n = 12

  Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

- vector<int> v, ans ;

  Traverse array and store the negative value

  v = { -4, -3, -9, -4, -2, -4 }

  Sort the Array

  v = { -9, -4, -4, -4, -3, -2 }

  Traverse from behind

  o/p: { -2, 2, }

map<int, int> mp;

1 -> 1

2 -> 1

4 -> 2

11 -> 1

Hello world

HASHING DATA STRUCTURE

- 3rd solution

- Int n = 12

  Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

- vector<int> v, ans ;

  Traverse array and store the negative value

  v = { -4, -3, -9, -4, -2, -4 }

  Sort the Array

  v = { -9, -4, -4, -4, -3, -2 }

  Traverse from behind

  o/p: { -2, 2, -4, 4 }

map<int, int> mp;

1 -> 1

2 -> 2

4 -> 2

11 -> 1

Hello World

HASHING DATA STRUCTURE

□ 3rd solution

□ Int n = 12

Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

□ vector<int> v, ans ;

Traverse array and store the negative value

v = { -4, -3, -9, -4, -2, -4 }

Sort the Array

v = { -9, -4, -4, -4, -3, -2 }

Traverse from behind

o/p: { -2, 2, -4, 4 }

map<int, int> mp;

1 -> 1

2 -> 2

4 -> 1

11 -> 1

Hello World

HASHING DATA STRUCTURE

□ **3rd solution**

□ **Int n = 12**

Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

□ **vector<int> v, ans ;**

Traverse array and store the negative value

v = { -4, -3, -9, -4, -2, -4 }

Sort the Array

v = { -9, -4, -4, -4, -3, -2 }

Traverse from behind

o/p: { -2, 2, -4, 4 , -4, 4}

map<int, int> mp;

1 -> 1

2 -> 2

4 -> 1

11 -> 1

Hello World

HASHING DATA STRUCTURE

- 3rd solution

- Int n = 12

  Arr[ ] = { 11, -4, 4, -3, -9, 4, -4, -2, 1, 2, -4, 2 }

- vector<int> v, ans ;

  Traverse array and store the negative value

  v = { -4, -3, -9, -4, -2, -4 }

  Sort the Array

  v = { -9, -4, -4, -4, -3, -2 }

  Traverse from behind

  o/p: { -2, 2, -4, 4 , -4, 4}

map<int, int> mp;

1 -> 1

2 -> 2

4 -> 0

11 -> 1

Hello World