**Pair with equal sum in C++**

```
#include <iostream>
#include <unordered_set>
#include <vector>

using namespace std;

bool sol(vector<int>& arr) {
    unordered_set<int> set;

    for (int i = 0; i < arr.size(); i++) {
        for (int j = i + 1; j < arr.size(); j++) {
            int sum = arr[i] + arr[j];
            if (set.count(sum)) {
                return true;
            } else {
                set.insert(sum);
            }
        }
    }
    return false;
}

int main() {
    vector<int> arr = {2, 9, 3, 5, 8, 6, 4};
    bool ans = sol(arr);
    cout << boolalpha << ans << endl;
    return 0;
}
```

## Input

```
arr = {2, 9, 3, 5, 8, 6, 4}
```

## Dry Run Table

| i | j | arr[i] | arr[j] | sum | Seen Sums Before | Is sum already in set? | Action |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 9 | 11 | {} | No | Insert 11 |
| 0 | 2 | 2 | 3 | 5 | {11} | No | Insert 5 |
| 0 | 3 | 2 | 5 | 7 | {11, 5} | No | Insert 7 |
| 0 | 4 | 2 | 8 | 10 | {11, 5, 7} | No | Insert 10 |
| 0 | 5 | 2 | 6 | 8 | {11, 5, 7, 10} | No | Insert 8 |
| 0 | 6 | 2 | 4 | 6 | {5, 7, 8, 10, 11} | No | Insert 6 |
| 1 | 2 | 9 | 3 | 12 | {5, 6, 7, 8, 10, 11} | No | Insert 12 |
| 1 | 3 | 9 | 5 | 14 | ... | No | Insert 14 |
| 1 | 4 | 9 | 8 | 17 | ... | No | Insert 17 |
| 1 | 5 | 9 | 6 | 15 | ... | No | Insert 15 |
| 1 | 6 | 9 | 4 | 13 | ... | No | Insert 13 |
| 2 | 3 | 3 | 5 | 8 | Already seen | ✅ **Yes** → Return `true` | |

## ✅ Output

```
true
```

Output:-
true