### Java

```java
class Solution {
  public int largestValsFromLabels(int[] values, int[] labels, int numWanted, int useLimit) {

  }
}
```

---

### JavaScript

```javascript
/**
 * @param {number[]} values
 * @param {number[]} labels
 * @param {number} numWanted
 * @param {number} useLimit
 * @return {number}
 */
var largestValsFromLabels = function(values, labels, numWanted, useLimit) {

};
```

---

### TypeScript

```typescript
function largestValsFromLabels(values: number[], labels: number[], numWanted: number, useLimit: number): number {

};
```

---

### C++

```cpp
class Solution {
public:
  int largestValsFromLabels(vector<int>& values, vector<int>& labels, int numWanted, int useLimit) {
  }
};
```

---

## C#

```
public class Solution {
  public int LargestValsFromLabels(int[] values, int[] labels, int numWanted, int useLimit) {

  }
}
```

---

## Kotlin

```
class Solution {
  fun largestValsFromLabels(values: IntArray, labels: IntArray, numWanted: Int, useLimit: Int): Int {

  }
}
```

---

## Go

```
func largestValsFromLabels(values []int, labels []int, numWanted int, useLimit int) int {

}
```

---