Java

```java
class Solution {
    public List<Integer> survivedRobotsHealths(int[] positions, int[] healths, String directions) {

    }
}
```

JavaScript

```javascript
/**
 * @param {number[]} positions
 * @param {number[]} healths
 * @param {string} directions
 * @return {number[]}
 */
var survivedRobotsHealths = function(positions, healths, directions) {

};
```

TypeScript

```typescript
function survivedRobotsHealths(positions: number[], healths: number[], directions: string): number[] {

};
```

C++

```cpp
class Solution {
public:
    vector<int> survivedRobotsHealths(vector<int>& positions, vector<int>& healths, string directions) {

    }
};
```

C#

```cs
public class Solution {
    public IList<int> SurvivedRobotsHealths(int[] positions, int[] healths, string directions) {

    }
}
```

Kotlin

```kt
class Solution {
    fun survivedRobotsHealths(positions: IntArray, healths: IntArray, directions: String): List<Int> {

    }
}
```

Go

```go
func survivedRobotsHealths(positions []int, healths []int, directions string) []int {

}
```
