2751. Robot Collisions

```
/*
    Time complexity: O(n log n)
    Extra space complexity: O(n)
*/
class Solution {
      public:
        class Robot{
            public:
                int index;
                int position;
                int health;
                char direction;
            public:
                Robot(int position,int health,char direction,int index): position(position),health(health),direction(direction),index(index){}
        };
    public:
        std::vector<int> survivedRobotsHealths(std::vector<int>& positions, std::vector<int>& healths, std::string directions){
            int n=positions.size();
            std::vector<Robot> robots;
            for(int i=0;i<n;++i){
               robots.push_back(Robot(positions[i],healths[i],directions[i],i));
            }

            std::sort(robots.begin(),robots.end(),[](const Robot& r1,const Robot& r2){return r1.position<r2.position;});
            std::stack<Robot> st;
            for(auto& robot: robots){
                    if(robot.direction=='R') st.push(robot);
                    else {
                        while(!st.empty() && healths[robot.index]>0){
                            if(robot.health==st.top().health){
                                healths[st.top().index]=0;
                                healths[robot.index]=0;
                                st.pop();

                            }
                            else if(robot.health<st.top().health) {
                                healths[st.top().index]--;
                                st.top().health--;
                                healths[robot.index]=0;
                            }
                            else {
                                healths[st.top().index]=0;
                                healths[robot.index]--;
                                robot.health--;
                                st.pop();
                            }
                        }
                    }
            }

            std::vector<int> ans;
            for(auto& h: healths){
                if(h!=0) ans.push_back(h);
            }

            return ans;

        }
};
```