

Mini-Max Sum

```
#include <bits/stdc++.h>

using namespace std;

string ltrim(const string &);
string rtrim(const string &);
vector<string> split(const string &);

/*
 * Complete the 'miniMaxSum' function below.
 *
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

void miniMaxSum(vector<int> arr) {
    sort(arr.begin(),arr.end());
    long long mini=0;
    long long maxi=0;
    long long tot=0;
    for(int i=0;i<arr.size();i++){
        tot+=arr[i];
    }
    mini=tot-arr[arr.size()-1];
    maxi=tot-arr[0];
    cout<<mini<<" "<<maxi;
}

int main()
{

    string arr_temp_temp;
    getline(cin, arr_temp_temp);

    vector<string> arr_temp = split(rtrim(arr_temp_temp));

    vector<int> arr(5);

    for (int i = 0; i < 5; i++) {
        int arr_item = stoi(arr_temp[i]);

        arr[i] = arr_item;
    }
```

```

        miniMaxSum(arr);

        return 0;
    }

string ltrim(const string &str) {
    string s(str);

    s.erase(
        s.begin(),
        find_if(s.begin(), s.end(), not1(ptr_fun<int,
int>(isspace)))
    );

    return s;
}

string rtrim(const string &str) {
    string s(str);

    s.erase(
        find_if(s.rbegin(), s.rend(), not1(ptr_fun<int,
int>(isspace))).base(),
        s.end()
    );

    return s;
}

vector<string> split(const string &str) {
    vector<string> tokens;

    string::size_type start = 0;
    string::size_type end = 0;

    while ((end = str.find(" ", start)) != string::npos) {
        tokens.push_back(str.substr(start, end - start));

        start = end + 1;
    }

    tokens.push_back(str.substr(start));

    return tokens;
}

```

