vector<string> split\_string(string);

// Complete the aVeryBigSum function below.

long aVeryBigSum(vector<long> ar) {

}

int main()

{

ofstream fout(getenv("OUTPUT\_PATH"));

int ar\_count;

cin >> ar\_count;

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

string ar\_temp\_temp;

getline(cin, ar\_temp\_temp);

vector<string> ar\_temp = split\_string(ar\_temp\_temp);

vector<long> ar(ar\_count);

for (int i = 0; i < ar\_count; i++) {

long ar\_item = stol(ar\_temp[i]);

ar[i] = ar\_item;

}

long result = aVeryBigSum(ar);

fout << result << "\n";

fout.close();

return 0;

}

vector<string> split\_string(string input\_string) {

string::iterator new\_end = unique(input\_string.begin(), input\_string.end(), [] (const char &x, const char &y) {

return x == y and x == ' ';

});

input\_string.erase(new\_end, input\_string.end());

while (input\_string[input\_string.length() - 1] == ' ') {

input\_string.pop\_back();

}

vector<string> splits;

char delimiter = ' ';

size\_t i = 0;

size\_t pos = input\_string.find(delimiter);

while (pos != string::npos) {

splits.push\_back(input\_string.substr(i, pos - i));

i = pos + 1;

pos = input\_string.find(delimiter, i);

}

splits.push\_back(input\_string.substr(i, min(pos, input\_string.length()) - i + 1));

return splits;

}