Question 1 : Write a program for concatenation of an array.

Answer :

```
class Solution {
public:
vector<int> getConcatenation(vector<int>& nums) {
int len = nums.size();
vector<int> ans(2*len, 0);
for(int i = 0; i<len; i++){
ans[i] = nums[i];
}
for(int i = 0; i < len; i++){
ans[i+len] = nums[i];
}
return ans;
}
};
```

Question 2 : Write a program to find running sum of an array.
Example: Input: nums = [1,2,3,4]
Output: [1,3,6,10]
Explanation: Running sum is obtained as follows: [1, 1+2, 1+2+3, 1+2+3+4].

Answer :

```
#include <iostream>

void running_sum(int arr[], int size) {
    for (int i = 1; i < size; i++) {
        arr[i] += arr[i - 1];
```

```
    }
}

int main() {
    int nums[] = {1, 2, 3, 4};
    int size = sizeof(nums) / sizeof(nums[0]);
    running_sum(nums, size);
    for (int i = 0; i < size; i++) {
        std::cout << nums[i] << " ";
    }
    return 0;
}
```

Question 3 : Write a program to find how many numbers are smaller than the current number and return ans in array form.
Example: Input: nums = [8,1,2,2,3]
Output: [4,0,1,1,3]
Explanation:
For nums[0]=8 there exist four smaller numbers than it (1, 2, 2 and 3).
For nums[1]=1 does not exist any smaller number than it.
For nums[2]=2 there exist one smaller number than it (1).
For nums[3]=2 there exist one smaller number than it (1).
For nums[4]=3 there exist three smaller numbers than it (1, 2 and 2).

Answer :

```
#include <iostream>
#include <algorithm>
#include<vector>
using namespace std;
vector<int> smaller_numbers(vector<int>& nums) {
    vector<int> result(nums.size());
    vector<int> sorted = nums;
```

```
    sort(sorted.begin(), sorted.end());
    for (int i = 0; i < nums.size(); i++) {
        result[i] = lower_bound(sorted.begin(), sorted.end(), nums[i]) - sorted.begin();
    }
    return result;
}

int main() {
    vector<int> nums = {8, 1, 2, 2, 3};
    vector<int> result = smaller_numbers(nums);
    for (int i = 0; i < result.size(); i++) {
        cout << result[i] << " ";
    }
    return 0;
}
```

Question 4 : Write a program to find maximum number of words in a sentence.

Example: Input: sentences = ["alice and bob love leetcode", "i think so too", "this is great thanks very much"]
Output: 6
Explanation:
- The first sentence, "alice and bob love leetcode", has 5 words in total.
- The second sentence, "i think so too", has 4 words in total.
- The third sentence, "this is great thanks very much", has 6 words in total.
Thus, the maximum number of words in a single sentence comes from the third sentence, which has 6 words.

Answer :

```
#include <bits/stdc++.h>
using namespace std;
int max_words(vector<string> sentences)
{
```

```
    int max_words = 0;
    for (string sentence : sentences)
    {
        int word_count = 0;
        for (int i = 0; i < sentence.size();)
        {
            int j = sentence.find(' ', i);
            if (j == string::npos) {
                word_count++;
                break;
            }
            word_count++;
            i = j + 1;
        }
        max_words = max(max_words, word_count);
    }
    return max_words;
}

int main()
{
    vector<string> sentences = {
        "alice and bob love leetcode",
        "i think so too",
        "this is great thanks very much"};
```

```
    cout << max_words(sentences) << endl;
    return 0;
}
```