

Write a modular C/C++ code which will

1. Generate an array with 100 random numbers in the range of 1-10000
2. The code then computes/checks if the number is odd/even, prime, factorial, prefix sum, average (use function calling function concept).

Code:

```
#include <iostream>

#include <vector>

#include <cstdlib>

#include <ctime>

#include <cmath>

#include <numeric>

using namespace std;

vector<int> generateRandomArray(int size, int min, int max);

bool isOdd(int num);

bool isPrime(int num);

long long factorial(int num);

vector<int> prefixSum(const vector<int>& arr);

double average(const vector<int>& arr);


int main() {

    const int size = 100;

    const int min = 1;

    const int max = 10000;

    vector<int> arr = generateRandomArray(size, min, max);

    for (int num : arr) {

        cout << "Number: " << num;

        cout << " | isEvenOrOdd: " << (isOdd(num) ? "Even" : "Odd");

        cout << " | isPrimeOrNotPrime: " << (isPrime(num) ? "prime" : "Not prime");

        cout << " | Factorial: " << factorial(num) << endl;

    }

}
```

```

vector<int> prefix_sums = prefixSum(arr);

cout << "Prefix sums: ";

for (int sum : prefix_sums) {
    cout << sum << " ";
}

cout << endl;

double avg = average(arr);

cout << "Average: " << avg << endl;

return 0;
}

vector<int> generateRandomArray(int size, int min, int max) {
    vector<int> arr(size);

    srand(time(0));

    for (int& num : arr) {
        num = min + rand() % (max - min + 1);
    }

    return arr;
}

bool isOdd(int num) {
    if(num%2==0)
        return true;
    else
        return false;
}

bool isPrime(int num) {
    if (num <= 1) return false;

    for (int i = 2; i <= sqrt(num); ++i) {
        if (num % i == 0) return false;
    }

    return true;
}

```

```

long long factorial(int num) {
    if (num <= 1) return 1;

    return num * factorial(num - 1);
}

vector<int> prefixSum(const vector<int>& arr) {
    vector<int> prefix_sums(arr.size());

    partial_sum(arr.begin(), arr.end(), prefix_sums.begin());

    return prefix_sums;
}

double average(const vector<int>& arr) {
    return static_cast<double>(accumulate(arr.begin(), arr.end(), 0)) / arr.size();
}

```

Output:

```

skynet@Vaibhavi: /mnt/d/STe x + v
skynet@Vaibhavi: /mnt/d/STe IT/third Year/CA$ g++ -pg RandomArrGen.cpp -o RandomArrGen
skynet@Vaibhavi: /mnt/d/STe IT/third Year/CA$ ./RandomArrGen
Number: 6869 | isEvenOrOdd: Odd | isPrimeOrNotPrime: prime | Factorial: 0
Number: 8830 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 7403 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 1392 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 3684 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 6515 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 8922 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 6574 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 619 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 2329 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 2802 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 8990 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 7142 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 574 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 917 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 419 | isEvenOrOdd: Odd | isPrimeOrNotPrime: prime | Factorial: 0
Number: 3101 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 8789 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 7943 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 1470 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 5381 | isEvenOrOdd: Odd | isPrimeOrNotPrime: prime | Factorial: 0
Number: 7877 | isEvenOrOdd: Odd | isPrimeOrNotPrime: prime | Factorial: 0
Number: 9320 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 4283 | isEvenOrOdd: Odd | isPrimeOrNotPrime: prime | Factorial: 0
Number: 6175 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 8899 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 1269 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 7852 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 3959 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 3392 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 8631 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 7179 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 2222 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 6033 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 8570 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 2257 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 2548 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 7492 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 8830 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0

```

```

skynet@Vaibhavi:/mnt/d/BTe x + v
Number: 1413 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 6194 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 3149 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 6594 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 5702 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 5672 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 3777 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 9652 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 8984 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 7885 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 4519 | isEvenOrOdd: Odd | isPrimeOrNotPrime: prime | Factorial: 0
Number: 9666 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 5993 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 4437 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 3443 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 123 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 9735 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 5696 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 4472 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 9315 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 4123 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 4071 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 2859 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 402 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 3981 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 3442 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 1664 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 4717 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 8950 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 8958 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 374 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 362 | isEvenOrOdd: Even | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 5151 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Number: 9875 | isEvenOrOdd: Odd | isPrimeOrNotPrime: Not prime | Factorial: 0
Prefix sums: 6869 15699 23102 24494 28178 34693 43615 50189 50799 53128 55130 63220 70362 70936 71853 72272 75373 84073 92016 93486 98867 106744 116064 1203
47 126522 135421 136681 144533 148492 151884 160515 167694 169916 175949 184519 186776 189324 196816 205646 215155 221327 228510 236109 239422 247179 252046
255777 262986 266553 274578 283256 288555 294457 298807 308388 316816 326416 333608 339888 343446 344030 348940 349677 358834 366129 371787 373200 379394 3
82543 389137 394839 400511 404288 413940 422924 430809 435328 444394 449487 453924 457367 457490 467225 472921 477393 486708 490831 494902 497761 498163 502
144 505586 507250 511967 520917 529875 530249 530611 535762 545637
Average: 5456.37
skynet@Vaibhavi:/mnt/d/BTech IT/third Year/CA$ gprof RandomArrGen gmon.out > sample.txt

```