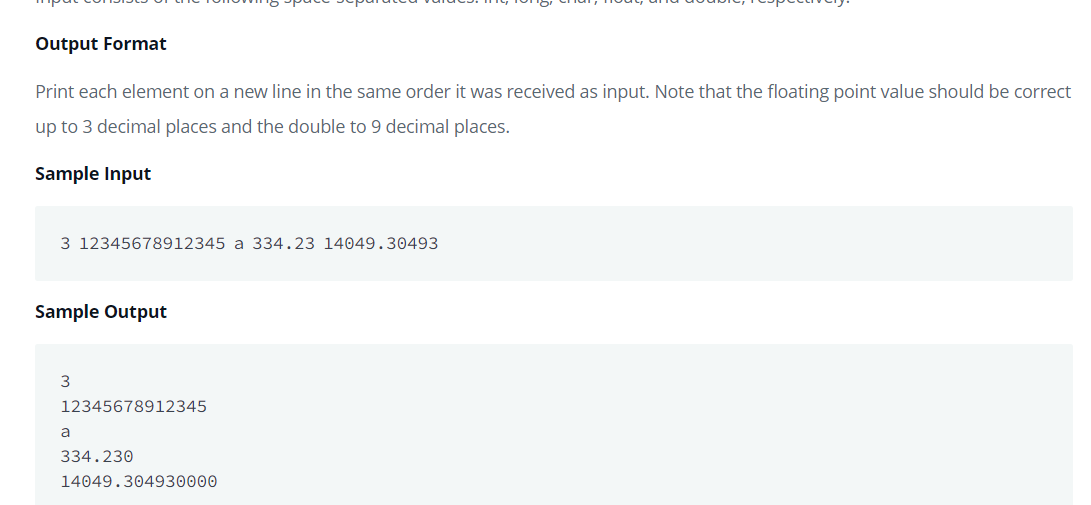
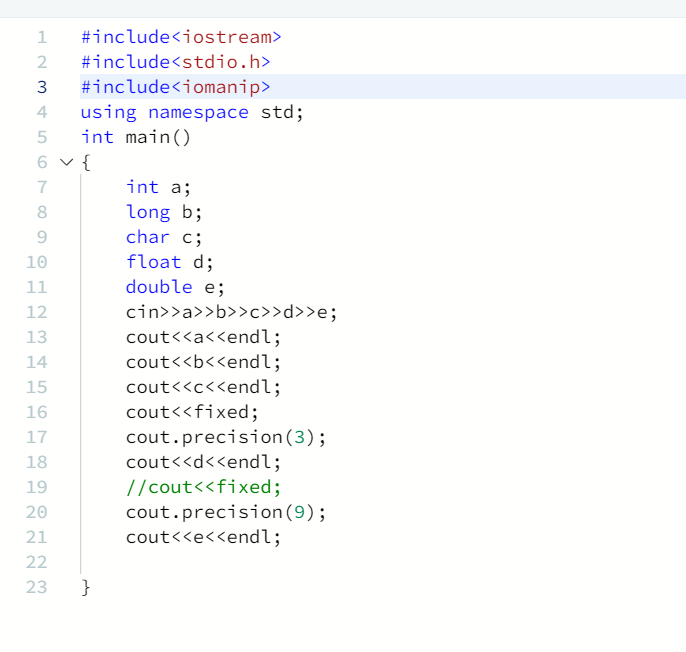
Date -29th September,2020

Question – to set the precision of the float number

Sample output:

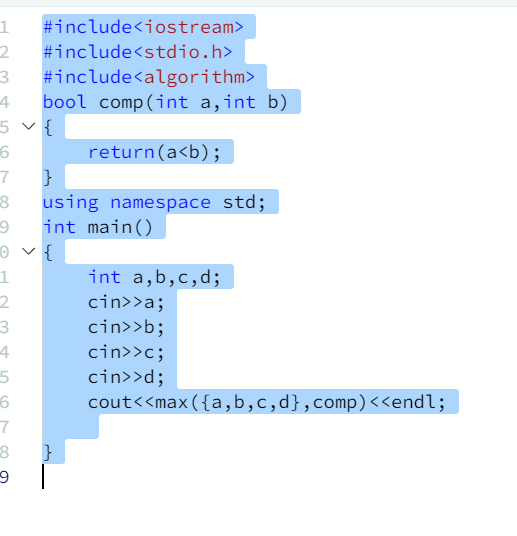


Consequent Code:



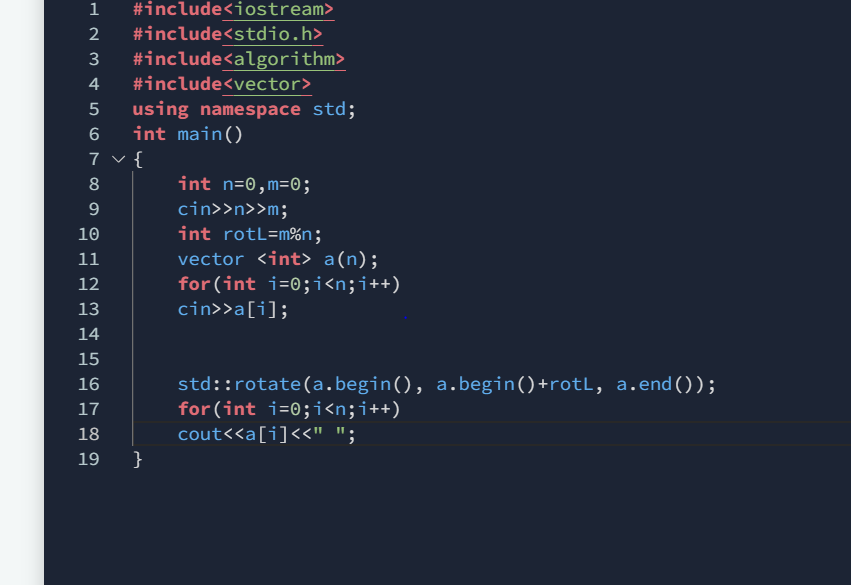
QUESTION: Innovative way to find the max of an array of elements

Code:

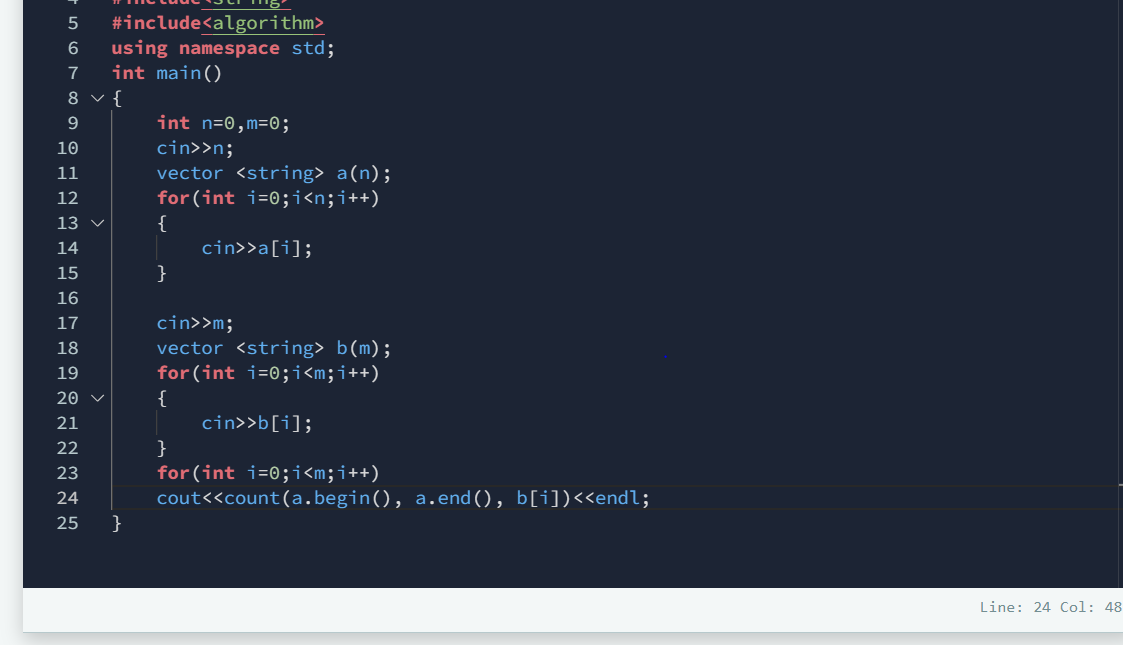


Question 3: How to left rotate the vector?

Code:

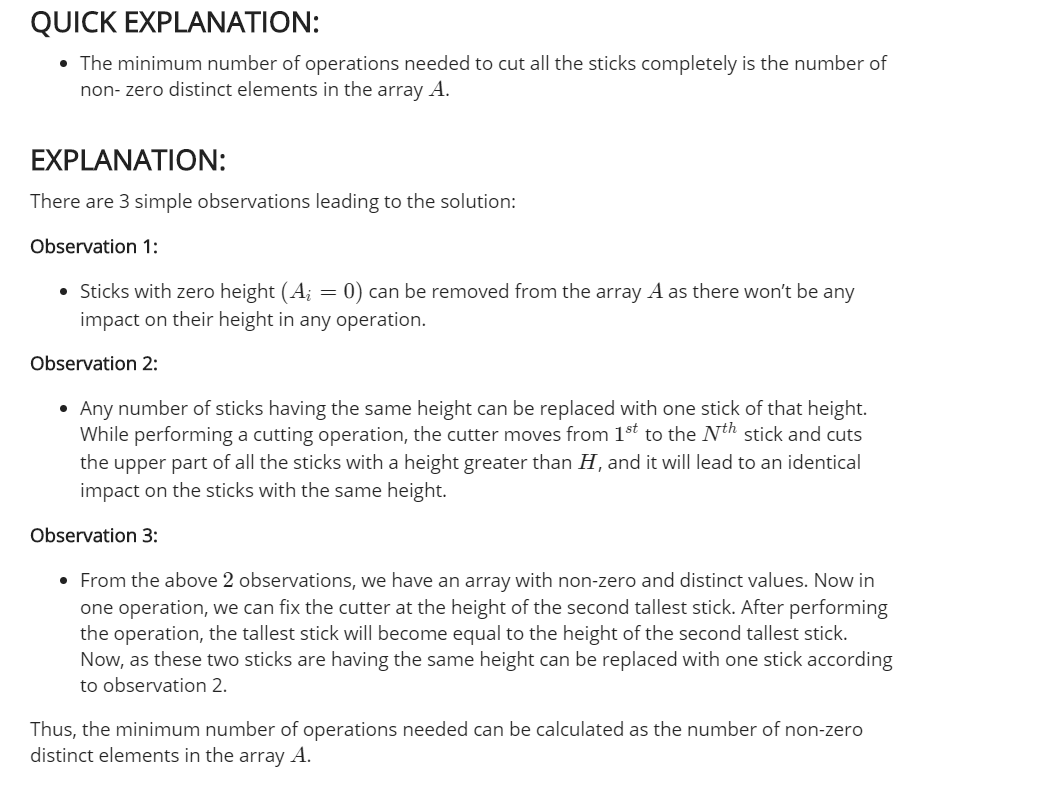


Question – check the string in the string vector:

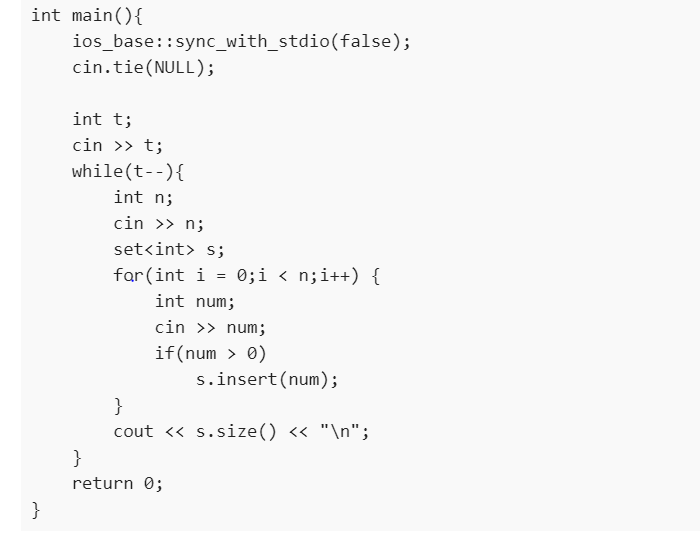


Question 5: The problem is that of cutting the sticks until its zero. So we need to find the minimum number of steps to cut all the sticks to zero.

Explanation:



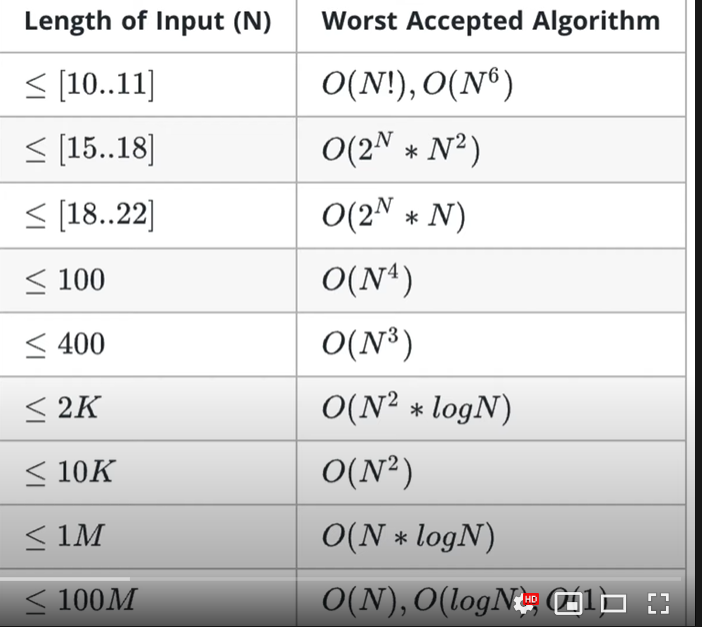
Code: Implementing it using sets in cpp



What I learnt in this approach is that sets can be used to input only the non distinct elements. So this helps in the above problem and we can find the shortest number of steps.

Questionn 6:Time Complexity for codechef.

Answer:



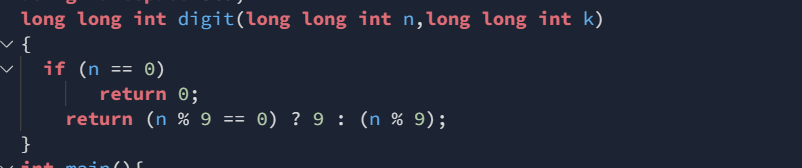
Ways to convert string to int-

Stoi- for string to integer

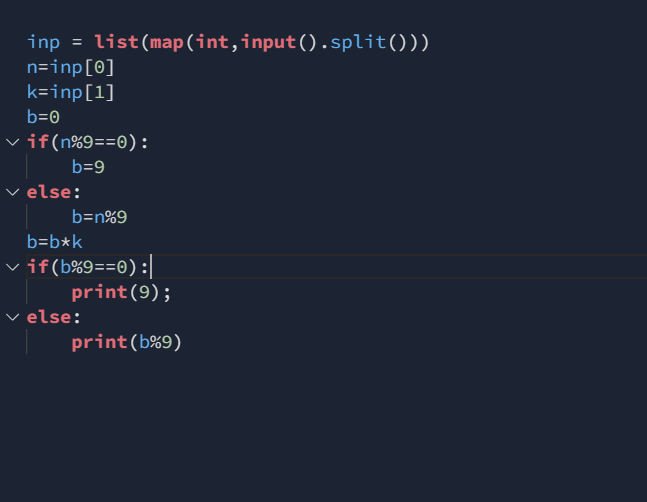
Stol- from string to long integer

Stoll- from string to long long integer

Question – find the digit sum – shortcut method-

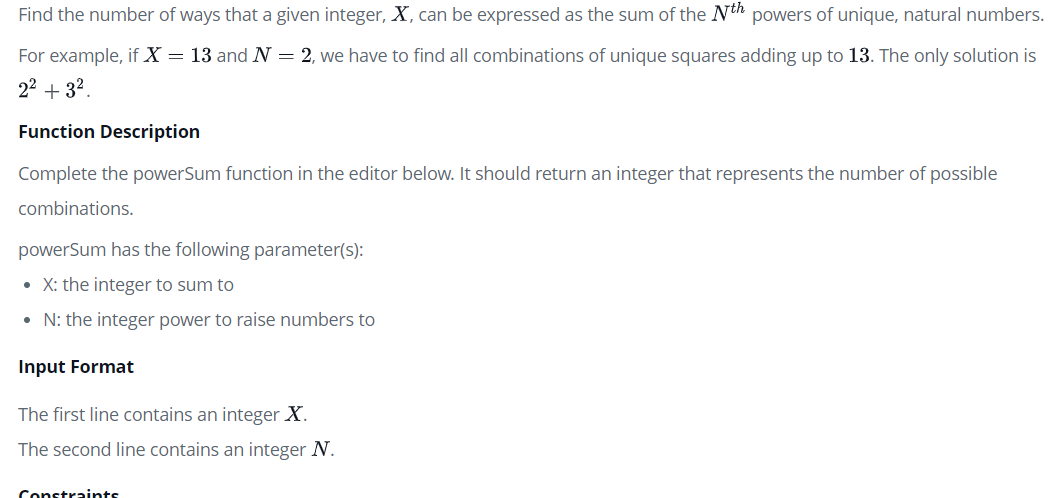


Question – Easiest way to find the digitsum and each number is appended k times:

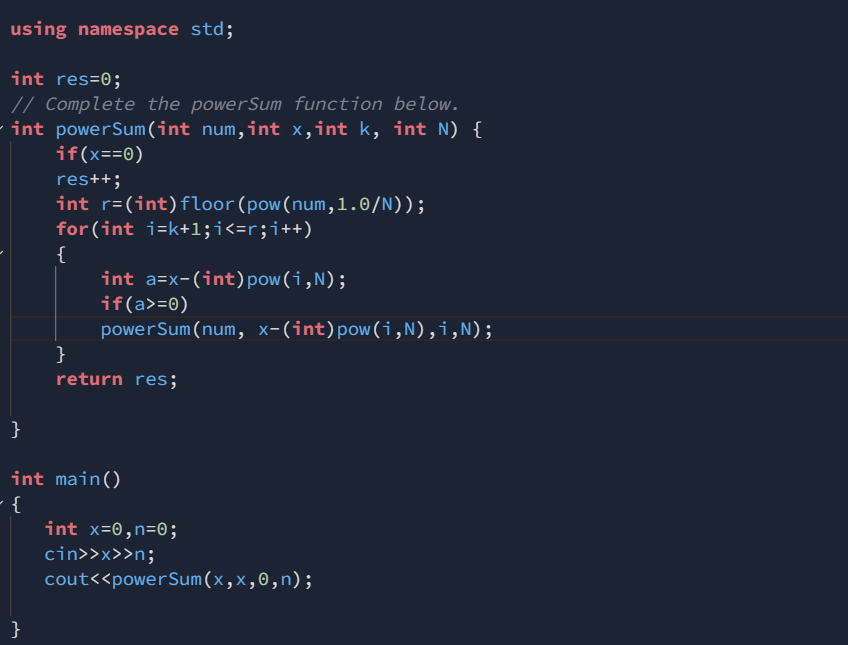
Answer: 

Above is the python code, I need to implement the same for very large numbers:

Question – To find the power sum ie:



Output:



Question – The key to store very big integers in the cpp is:

#include<iostream>

using namespace std;

#define MAX 500

int multiply(int x, int res[], int res\_size);

void factorial(int n)

{

int res[MAX];

res[0] = 1;

int res\_size = 1;

for (int x=2; x<=n; x++)

res\_size = multiply(x, res, res\_size);

cout << "Factorial of given number is \n";

for (int i=res\_size-1; i>=0; i--)

cout << res[i];

}

int multiply(int x, int res[], int res\_size)

{

int carry = 0;

for (int i=0; i<res\_size; i++)

{

int prod = res[i] \* x + carry;

res[i] = prod % 10;

carry = prod/10;

}

while (carry)

{

res[res\_size] = carry%10;

carry = carry/10;

res\_size++;

}

return res\_size;

}

int main()

{

factorial(100);

return 0;

}

Question – Faster way to check if a number is perfect or not:

bool isPerfect(long long int n)

{

// To store sum of divisors

long long int sum = 1;

// Find all divisors and add them

for (long long int i=2; i\*i<=n; i++)

{

if (n%i==0)

{

if(i\*i!=n)

sum = sum + i + n/i;

else

sum=sum+i;

}

}

// If sum of divisors is equal to

// n, then n is a perfect number

if (sum == n && n != 1)

return true;

return false; }

Question- Faster way to check if a number is prime or not

Code:

int isPrime(int n)

{

int i, flag = 1;

// Ask user for input

//printf("Enter a number: \n");

// Store input number in a variable

//scanf("%d", &n);

// Iterate from 2 to n/2

for (i = 2; i <= sqrt(n); i++) {

// If n is divisible by any number between

// 2 and n/2, it is not prime

if (n % i == 0) {

flag = 0;

break;

}

}

if(n<=1)

flag=0;

else if(n==2)

flag=1;

return flag;

}

Question – Recursive approach to sort a vector array

Answer –

#include<iostream>

#include<vector>

#include<algorithm>

using namespace std;

void insert(vector <int> &a,int temp)

{

if(a.size()==0 || a[a.size()-1]<=temp)

{

a.push\_back(temp);

return ;

}

int val=a[a.size()-1];

a.pop\_back();

insert(a,temp);

a.push\_back(val);

return ;

}

void sorty(vector <int> &a)

{

if(a.size()==1)

return;

int temp=a[a.size()-1];

a.pop\_back();

sorty(a);

insert(a,temp);

}

int main()

{

int n=0;

cout<<"Enter the length of the vector :";

cin>>n;

vector <int> a(n);

cout<<"Enter the vector elements "<<endl;

for(int i=0;i<n;i++)

{

cin>>a[i];

}

sorty(a);

for(int i=0;i<n;i++)

{

cout<<a[i]<<" ";

}

}

Question :Return Two numbers in cpp from a function:

