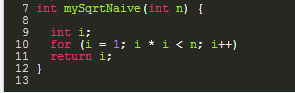
Debugging

1. Input : sebuah bilangan kuadrat

output : akar dari input



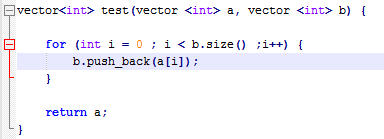
: for (i = 1; i \* i < n; i++) diubah menjadi **for (i = 1; i \* i <= n; i++);**

1. Input 1 : Vector <int>

input 2 : Vector <int>

output : sebuah array vektor dengan elemen dari input 1 diikuti oleh input 2

cth : Input 1 {2,2} , Input 2 {3,3,1} , output {2,2,3,3,1}

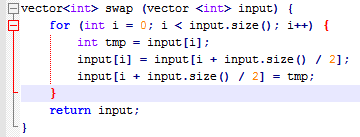


: **b.push\_back(a[i]);** diganti **a.push\_back(b[i]);**

1. Input : vector<int> dengan panjang Data L (L pasti genap)

Output: vector<int> Dengan L/2 data terakhir diikuti L/2 data pertama

cth: Input [1, 3, 2, 1] , maka output [2, 1, 1, 3]



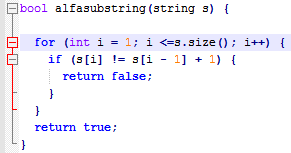
: input.size() / 2

1. "alfabet" adalah sebuah string yang terdiri dari karakter "abcde....xyz"

Input : sebuah string

output : "true" jika string input merupakan substring dari alfabet, "false" sebaliknya

cth : "efghi" --> true , sementara , "bde" --> false



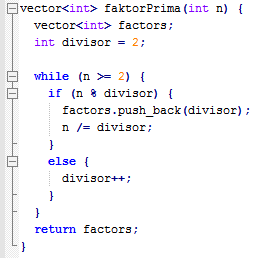
For int i diuabah menajdi **i = 1; i< s.size();**

1. Sebuah fungsi yang menghasilkan vector berisi semua faktor prima sebuah bilangan

Input : positif integer

Output: vector yg berisi faktor prima dari input yang terurut dari kecil ke besar

Cth : input = 100 , output = {2,2,5,5}



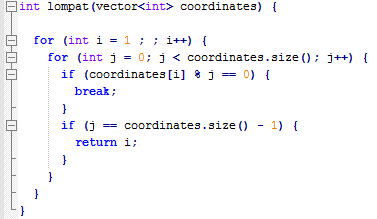
: line 4 diubah ke **if (n % divisor == 0 ) {**

1. Input : Vector yang berisi angka2 positif yang merepresentasikan koordinat dari paku pada sebuah garis lurus (sumbu x)

Anda diminta untuk melompat2 pada garis lurus yg berpaku tersebut , tetapi lompatan yang anda lakukan harus konstan panjangnya,

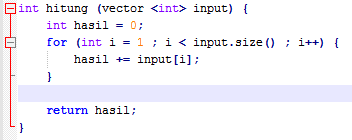
Output : minimal panjang lompatan agar gak terkena paku dari awal sampe akhir, asumsikan akhir adalah setelah melewati semua paku,

cth : Input = {5,3,6,9,7} maka output = 4



1. Input : Vector <int> yang tidak kosong

output : jumlahan semua elemen pada Input



: for diubah menjadi **for (int i = 0 ; i < input.size() ; i++) {**

1. Fungsi yang terdiri dari 3 parameter, 2 parameter di pastikan sama, dan 1 lagi yang berbeda sendiri,

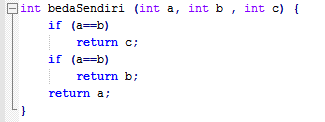
Input 1 : integer

Input 2 : integer

Input 3 : integer

output : parameter yang berbeda sendiri

cth : Input = {2,4,2} , maka output = 4

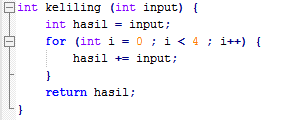


: if ( a==b ) diubah mejadi **if ( a==c );**

1. Input : sebuah integer

Output: keliling dari kubus dengan panjang sisi sebesar input

cth : input = 1 , maka output = 4

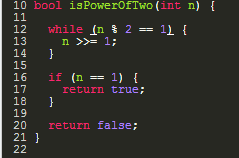


: int hasil = input; diubah menjadi **int hasil = 0;**

1. Input : positif integer

Output : "true" jika Input adalah bilangan 2 pangkat , "false" jika bukan

cth: input = 2 , maka output = True || Input = 3 , maka output = False

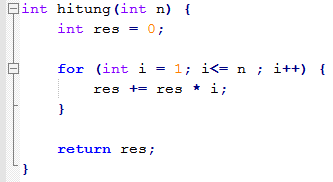


: **while( n%2 == 0 ) [**

1. Input : positif integer

output : jumlah dari 12 , 22 , .... , sampai (Input)2

cth : Input = 5, maka output = 55

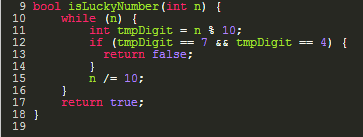


: res += res \* i ; diubah menjadi **res += i \* I;**

1. "angka keberuntungan" adalah angka positif yang setiap digitnya hanya terdiri dari digit 4 dan 7

Input : positif integer

Output : "true" jika Input merupakan angka keberuntungan, "false" jika sebaliknya



: if dibaris ke 4 diubah menjadi **if ( tmpDigit != 7 || tmpDigit != 4 ) {**

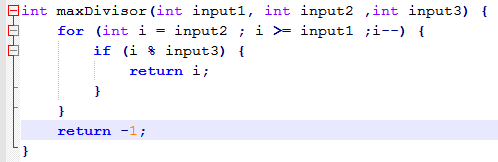
1. Input 1 : integer

input 2 : integer (input 2 >= Input 1)

input 3 : positif integer (>= 0)

output : maksimal angka dari input 1 sampai input 2 yang habis dibagi oleh input 3

, outputkan -1 jika tidak ada jawaban.

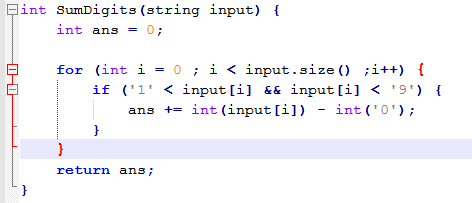


: if di baris ke 3 dibubah ke **if ( i % input3 == 0 ) {**

1. Input : sebuah string

output : jumlah dari semua digit yang muncul di string

cth : input = "2 mobil , 12 motor" , maka ouputnya = 5



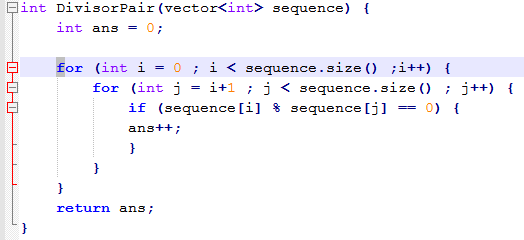
: if di barus ke 4 ( 5 ) diubah menjadi **if ( ‘1’ <= input[i] && input[i] <= ‘9’ ) {**

1. Input : vector<int> yang isinya unik dan terurut menaik.

output : jumlah dari pasangan terurut dari elemen pada input sehingga 1 elemen pada pasangan habis membagi pasangannya .

cth : input = {1,2,3} --> maka output = 2 ---> {1,2} dan {1,3}

input = {2,4,8} --> maka output = 3 ---> {2,4} , {2,8} dan {4,8}



:

: line ke 6 ( 5 ) diganti **if ( sequence[j] % sequence[i] == 0 ) {**