Chiara capocello

PROBLEMA 1

//COMPLESSITà O(nlogn)

#include <iostream>

using namespace std;

#define N 1000

// funzione che restituisce il numero di occorrenze di k (elemento che sto cercando)

int conteggio(int v[N], int first, int n, int k)

{

// caso base -> quando c'è solo un elemento, se è uguale a quello che sto

// cercando lo restituisco altrimenti restituisco 0

if (first == n)

{

if (v[first] == k)

{

return 1;

}

else

return 0;

}

else

{

// ricorsione

int m = (first + n) / 2;

int count\_1 = conteggio(v, first, m, k);

int count\_2 = conteggio(v, m + 1, n, k);

return (count\_1 + count\_2);

}

}

int main()

{

int test;

cin >> test;

while (test--)

{

int k; // numero da cercare

int v[N];

int n; // dim array

cin >> k;

cin >> n;

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

{

cin >> v[i];

}

cout << conteggio(v, 0, n - 1, k);

}

return 0;

}

PROBLEMA 2

#include <iostream>

#include <vector>

#include <cmath>

using namespace std;

vector<int> primi;

vector<int> sequenza;

void stampa\_sequenza(vector<int> sequenza)

{

for (int i = 0; i < sequenza.size(); i++)

{

cout << sequenza[i] << " ";

}

cout << endl;

}

bool verifica\_primo(int primo)

{

// verifica radice quadrata

int radice = sqrt(primo);

// poichè 1 non è sicuramente un numero primo

if (primo == 1)

{

return false;

}

// verifico se è divisibile e ritorno true solo se è vero cioe se è un numero priimo

for (int i = 2; i <= radice; i++)

{

if (primo % i == 0)

return false;

}

return true;

}

void somma(int totale, int N, int S, int i)

{

// caso base

if (totale == S && sequenza.size() == N)

{

stampa\_sequenza(sequenza);

return;

}

if (totale > S || i == primi.size())

{

return;

}

sequenza.push\_back(primi[i]);

somma(totale + primi[i], N, S, i + 1);

// BACKTRACK-> rimuovo l'elemento dal vettore sequenza

sequenza.pop\_back();

somma(totale, N, S, i + 1);

}

void sequenza\_numeri\_primi(int S, int N, int P)

{ // mi genera tutti i numeri minori della somma partendo da P+1, verifica se sono primi

for (int i = P + 1; i <= S; i++)

{

if (verifica\_primo(i))

primi.push\_back(i);

}

if (primi.size() < N)

return;

somma(0, N, S, 0);

}

int main()

{

int test;

cin >> test;

int i = 0;

while (test--)

{

int S; // indica la somma

int N; // lunghezza della sequenza

int P; // numero primo

cin >> S;

cin >> N;

cin >> P;

i++;

cout << "CASO DI TEST " << i << endl;

sequenza\_numeri\_primi(S, N, P);

}

cout << " fine test";

return 0;

}