

Generare tablou cu valori distincte

Programul nu mai este legat de cautari, deci va invit pe toti sa incercati sa il faceti, chiar daca nu ati fost "pe faza" la programele anterioare. Este un capitol nou, este prima tema din mai multe care se vor adauga la aceasta.

Scrieti un program care sa genereze un tablou cu primele N numere naturale (N este o constanta intreaga pozitiva ce poate fi 8 sau mai mult), distincte intre ele (niciun numar nu apare de doua ori), permutate aleatoriu. Chiar daca N este 100, programul ar trebui sa ruleze in mai putin de o secunda.

Exemplu:

Daca N este 8, atunci un rezultat ar putea fi:

5 7 1 4 8 2 6 3

→ Sunt afisate toate numerele de la 1 la 8 si sunt permutate aleatoriu.

Aceasta reprezentare ne va ajuta ulterior sa construim o reprezentare a problemei aranjarii damelor sau pentru comis-voiajor. Vom continua cu o rezolvare folosind Hill Climbing. Vedeti aici: <http://inf.ucv.ro/documents/cstoean/c5IA.pdf>, incepand cu slide-ul 41.

Soluții

Am ales cele mai economice soluții pe care le-am primit

Soluția 1

Observații: o soluție simplă și eficientă. Spre deosebire de exemplul pe care îl sugeram eu, este chiar mai simplă, fiindcă nu face deplasarea elementelor în tabloul a pentru a șterge elementul de pe poziția g, ci îl înlocuiește direct cu elementul de pe ultima poziție din a.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <iostream>

#define N 8

using namespace std;

void permutare(int d[]) {
    int n=N, a[N];
    for (int i = 0; i < N; i++)
        a[i] = i + 1;

    while(n>0)
        for (int i = 0; i < N; i++)
        {
            int g = rand() % n;
            d[i] = a[g];
            a[g] = a[n - 1];
            n--;
        }
}

int main() {
    int d[N];
    srand(time(NULL));
    permutare(d);
    cout << "Numerele sunt:" << endl;
    for (int i = 0; i < N; i++)
        cout << d[i] << " ";
    return 0;
}
```

Programul compilat

```
Numerele sunt:
5 1 7 6 2 3 4 8
```

Soluția 2

Observație: programul care urmează întocmai instrucțiunile de pe slide-uri.

```

#include<iostream>
#include<time.h>
#include<stdlib.h>
#include<stdio.h>
#define N 8

using namespace std;

void initializare(int v[]) {
    int m[N], lungime;

    lungime = N;

    for (int j = 0; j < N; j++)
        m[j] = j+1;

    while (lungime > 0)
        for (int i = 0; i < N; i++)
        {

            int g = rand() % lungime;
            v[i] = m[g];

            for (int j = g; j < N; j++)
                m[j] = m[j + 1];

            lungime--;
        }
}

int main() {
    int v[N];
    srand(time(NULL));
    initializare(v);
    cout << "Numere permutate sunt: " << endl;
    for (int i = 0; i < N; i++)
        cout << v[i] << " ";
    return 0;
}

```

Programul compilat

```

Numere permutate sunt:
5 3 1 4 7 2 8 6

```

Soluția 3

Observație: Am primit de la un student două metode, ambele diferite de cele de mai sus.

```

//
// Metoda 1 ( Am folosit std::Vector, std::shuffle )
//

#include <iostream>
#include <vector>

```

```

#include <algorithm>
#include <random>

int main()
{
    std::vector<unsigned int>Numbers;

    for (int i = 0; i < 100; i++)
        Numbers.push_back(i);

    std::cout << "Seed = " << std::endl;
    std::shuffle(Numbers.begin(), Numbers.end(),
std::default_random_engine(4142020));

    for (const auto& i : Numbers)
        printf("%d ", i);

    return 0;
}

//
// Metoda 2 ( Am implementat algoritmul Fisher-Yates shuffle )
//
https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates\_shuffle#Fisher\_and\_Yates.27\_original\_method

#include <iostream>

int main()
{
    int arr[100];

    for (int i = 0; i < 100; i++)
        arr[i] = i;

    // Algoritm
    // To shuffle an array a of n elements(indices 0..n - 1) :
    //     for i from n - 1 downto 1 do
    //         j = random integer with 0 <= j <= i
    //         exchange a[j] and a[i]

    for (int i = 99; i > 0; i--)
    {
        int j = rand() % (i + 1);

        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }

    for (int i : arr)
        printf("%d, ", i);
}

```