

C++ - Losowanie liczb

© Copyright by 3bird Projects 2022, <http://edukacja.3bird.pl>

Ogólne

W wersji podstawowej zadania, losowany jest po prostu ciąg liczb. W wersji rozbudowanej, można te liczby posortować i policzyć, która z nich pojawiła się największą ilość razy.

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii i tzw. twarde odstępy. Kod należy przepisać ze zrozumieniem.

Kod - wersja podstawowa

```
#include <iostream>
```

```
using namespace std;
```

```
const int iloscElementow = 10;
```

```
const int gornyZakres = 9;
```

```
int numerElementuTablicy, tablicaLiczby[iloscElementow];
```

```
// ===== FUNKCJA "GENERUJ" =====
```

```
void generujLiczby() {
```

```
    system("cls");
```

```
    cout << " :: GENEROWANIE 10 LICZB LOSOWYCH:\n\n";
```

```
    // WYGENEROWANIE TABLICY Z LICZBAMI PSEUDOLOSOWYMI
```

```
    for (numerElementuTablicy=0; numerElementuTablicy<iloscElementow;
```

```
    numerElementuTablicy++) {
```

```
        cout << " " << (tablicaLiczby[numerElementuTablicy] = rand() % (gornyZakres+1))
```

```
        << " ";
```

```
    }
```

```
}
```

```
// ===== FUNKCJA "MAIN" =====
```

```
int main(void) {
```

```
    string decyzjaUzytkownikaGenerowanie = "Tak";
```

```
    while ((decyzjaUzytkownikaGenerowanie == "Tak") || (decyzjaUzytkownikaGenerowanie ==  
    "tak") || (decyzjaUzytkownikaGenerowanie == "t") || (decyzjaUzytkownikaGenerowanie ==  
    "T")) {
```

```
        generujLiczby();
```

```
        cout << "\n\nWygenerowac liczby ponownie?" << endl;
```

```
        cout << "\n\nTwoja odpowiedz (Tak/Nie): ";
```

```
        getline(cin, decyzjaUzytkownikaGenerowanie); // Pierwszy parametr przekazuje wartość  
        drugiemu (także pusty ENTER).
```

```
    }
```

```
    cout << "\n\nNaciśnij ENTER, aby zamknąć..." << endl;
```

```
system("PAUSE > nul");  
}
```

Kod - wersja rozbudowana

```
#include <iostream>  
#include <time.h> // Potrzebne do srand()  
#include <windows.h> // Potrzebne do Sleep()  
#include <algorithm> // Potrzebne do transform()  
using namespace std;  
  
const int iloscElementow = 50;  
const int gornyZakres = 9;  
int numerElementuTablicy, tablicaLiczb[iloscElementow], najczestszyElement, iloscWystapien,  
pozycjaElementuNajczestszego, licznikSprawdzen,j;  
  
// ===== FUNKCJA "GENERUJ" =====  
// Wyrażenie "void", bo funkcja nie musi niczego zwracać za pomocą "return", wykonuje po  
// prostu instrukcje:  
void generujLiczby() {  
    // Czyścimy ewentualne poprzednie losowania:  
    system("cls");  
    // Wprowadzamy ziarno (seed) oparte o aktualny czas, które zostanie zaraz wykorzystane  
    // przez funkcję rand():  
    srand(time(0));  
    cout << ":: GENEROWANIE 50 LICZB LOSOWYCH:\n\n";  
    // WYGENEROWANIE TABLICY Z LICZBAMI PSEUDOLOSOWYMI  
    for (numerElementuTablicy=0; numerElementuTablicy<iloscElementow;  
numerElementuTablicy++) {  
        // Funkcja rand() losuje ciągle te same liczby, podaje je jednak w różnej kolejności.  
        // Funkcja srand() wprowadza do funkcji rand() tzw. "ziarno" (seed).  
        // Z każdym przebiegiem pętli wkładana jest do tablicy jedna wylosowana liczba:  
        cout << " " << (tablicaLiczb[numerElementuTablicy] = rand() % (gornyZakres+1))  
        << " ";  
        // Opóźnienie w generowaniu poszczególnych liczb (ozdobnik wizualny):  
        Sleep(100);  
    }  
}  
  
// ===== FUNKCJA "WYSZUKAJ" =====  
// W przypadku, gdy użytkownik wybierze wyszukanie najpopularniejszej liczby, poprzedni wypis  
// jest czyszczony (cls) i generowany jest nowy, już wskazujący na najpopularniejszą liczbę (nie  
// można tego zrobić inaczej niż wypisanie nowej wersji tablicy).  
void wyszukajNajpopularniejsza() {  
    // Aby wyczyścić zmienne z poprzednich wartości / statystyk ustawiamy im początkowe  
    // wartości:  
    najczestszyElement = tablicaLiczb[0];  
    pozycjaElementuNajczestszego = 0;  
    iloscWystapien = 0;
```

```

// Pętla przejdzie po każdej wylosowanej liczbie. Weźmie pierwszą i sprawdzi ile razy się
// powtórzy (druga pętla). Innymi słowy: Pierwsza pętla FOR użyta jest, żeby wybrać
// pojedynczy element tablicy, a druga porównuje ten element z całą tablicą.
for (numerElementuTablicy=0; numerElementuTablicy<iloscElementow;
numerElementuTablicy++) {
    // Poniższy licznik będzie zliczał ilość wystąpień danej liczby w tablicy:
    licznikSprawdzen = 0;
    for (j=0; j<iloscElementow; j++) {
        if (tablicaLiczb[numerElementuTablicy] == tablicaLiczb[j]) {
            licznikSprawdzen++;
        }
    }
    // Jeśli pętla przeszła do końca i sprawdziła wszystkie liczby, to przypisz wartości do
    // zmiennych:
    if (licznikSprawdzen > iloscWystapien) {
        najczestszyElement = tablicaLiczb[numerElementuTablicy];
        pozycjaElementuNajczestszego = numerElementuTablicy;
        iloscWystapien = licznikSprawdzen;
    }
}
// Czyścimy ekran po wygenerowanych liczbach, bo teraz chcemy wyświetlić je jeszcze raz,
// ale już ze wskazaniem tych najbardziej popularnych:
system("cls");
cout << ":: GENEROWANIE 50 LICZB LOSOWYCH:\n\n";
// Wypisanie najpopularniejszej liczby w kwadratowym nawiasie:
for (numerElementuTablicy=0; numerElementuTablicy<iloscElementow;
numerElementuTablicy++) {
    if (tablicaLiczb[numerElementuTablicy] == najczestszyElement) {
        cout << " \033[1;31;40m[" << tablicaLiczb[numerElementuTablicy] << "]\033[0m";
    }
    else {
        cout << " " << tablicaLiczb[numerElementuTablicy] << " ";
    }
}
// Wypisanie statystyk:
cout << "\n\n\n:: WYNIKI WYSZUKIWANIA:" << endl;
cout << "Najczęściej występująca liczba: " << najczestszyElement << "." << endl;
cout << "Wystąpiła w ciągu: " << iloscWystapien << " razy." << endl;
cout << "Po raz pierwszy wystąpiła na pozycji nr: " << pozycjaElementuNajczestszego + 1
<< ".\n\n\n";
}

// ===== FUNKCJA "MAIN" =====
// Wyrażenie "void", bo funkcja nie pobiera żadnych parametrów.
int main(void) {
    string decyzjaUzytkownikaGenerowanie = "tak";

```

```
string decyzjaUzytkownikaWyszukiwanie = "tak";
```

```
while ((decyzjaUzytkownikaGenerowanie == "tak") || (decyzjaUzytkownikaGenerowanie == "t")) {  
    generujLiczby();  
    cout << "\n\nCzy mam wyszukać najpopularniejszą liczbę w tym ciągu?\n\nTwoja odpowiedź (Tak/Nie): ";  
    // Pierwszy parametr przekazuje wartość drugiemu (także pusty ENTER):  
    getline(cin, decyzjaUzytkownikaWyszukiwanie);  
    // Odpowiedź, którą wprowadził użytkownik, zamieniamy na małe litery, aby było o połowę  
    // mniej wariantów do sprawdzenia:  
    transform(decyzjaUzytkownikaWyszukiwanie.begin(),  
    decyzjaUzytkownikaWyszukiwanie.end(),  
    decyzjaUzytkownikaWyszukiwanie.begin(), ::tolower);  
    if ((decyzjaUzytkownikaWyszukiwanie == "tak") || (decyzjaUzytkownikaWyszukiwanie == "t")) {  
        wyszukajNajpopularniejsza();  
    }  
  
    cout << "Wygenerować liczby ponownie?" << endl;  
    cout << "Twoja odpowiedź (Tak/Nie): ";  
    // Pierwszy parametr przekazuje wartość drugiemu (także pusty ENTER):  
    getline(cin, decyzjaUzytkownikaGenerowanie);  
    // Odpowiedź, którą wprowadził użytkownik, zamieniamy na małe litery, aby było o połowę  
    // mniej wariantów do sprawdzenia:  
    transform(decyzjaUzytkownikaGenerowanie.begin(),  
    decyzjaUzytkownikaGenerowanie.end(),  
    decyzjaUzytkownikaGenerowanie.begin(), ::tolower);  
}  
cout << "\n\nNaciśnij ENTER, aby zamknąć..." << endl;  
system("pause > nul");  
}
```

Ostatnia aktualizacja: 26 grudnia 2022.