

Nazwisko i imię: **Soczyński Krzysztof**

Kierunek: **Informatyka w inżynierii komputerowej**

Rok: I Semestr: I Forma studiów: **niestacjonarne**

Grupa: **Lk4**

Przedmiot: **Podstawy programowania w C/C++**

Zajęcia: **Laboratorium komputerowe 1**

Data: **29.10.2022**

1. Treść zadania

Wygeneruj 10 liczb losowych, oblicz wartości funkcji $\sin()$ tych liczb, następnie wypisz wygenerowane liczby i wartości funkcji $\sin()$ w konsoli oraz do pliku.

2. Kod programu

```
#include <algorithm>
#include <array>
#include <cmath>
#include <fstream>
#include <iomanip>
#include <iostream>
#include <random>

int main() {
    constexpr size_t arraySize = 10;
    constexpr std::streamsize streamSize = 4;

    std::array<double, arraySize> numbers{};
    std::array<double, arraySize> sinusOfNumbers{};
    std::random_device randomDevice{};
    std::mt19937 randomEngine{ randomDevice() };
    std::uniform_real_distribution distribution(-100.0, 100.0);

    std::ranges::generate(numbers, [&distribution, &randomEngine]{
        return distribution(randomEngine);
    });
    std::ranges::generate(sinusOfNumbers, [i = 0, &numbers]() mutable {
        return std::sin(numbers[i++]);
    });

    std::fstream fileStream{ "file.txt", std::ios::out };

    for (size_t i{0}; i < arraySize; ++i) {
        const auto& n = numbers[i];
        const auto& sn = sinusOfNumbers[i];

        std::cout << std::setw(streamSize) << std::setprecision(streamSize) << n << '\t' << sn << '\n';
        fileStream << std::setw(streamSize) << std::setprecision(streamSize) << n << '\t' << sn << '\n';
    }
    std::cout << std::endl;
    fileStream << std::endl;

    return 0;
}
```

3. Opis programu

3.1. Dołączone biblioteki

- **<algorithm>**
zbiór funkcji przydatnych w różnych sytuacjach, np. sortowania, szukania, zliczania, manipulowania
- **<array>**
biblioteka implementująca kontener wchodzący w skład STL (Standard Template Library) będący odpowiednikiem statycznej tablicy
- **<cmath>**
biblioteka zawierająca funkcje matematyczne
- **<fstream>**
biblioteka dająca możliwość operowania na plikach
- **<iomanip>**
biblioteka pozwalająca manipulować strumieniami wejścia/wyjścia
- **<iostream>**
biblioteka pozwalająca wczytywać dane i wypisywać dane z/do strumienia wejścia/wyjścia
- **<random>**
dostępna od C++11 biblioteka pozwalająca generować liczby losowe i pseudo-losowe

3.2. Kod „linijka po linijce”

```
constexpr size_t arraySize = 10;  
constexpr std::streamsize streamSize = 4;
```

Inicjalizacja dwóch stałych (*arraySize* określa liczbę wygenerowanych liczb, *streamSize* zostanie użyte do określenia szerokości pola w strumieniach wyjścia).

constexpr oznacza, że w tym przypadku wartości zmiennych dedukowane będą w trakcie kompilacji.

```
std::array<double, arraySize> numbers{};  
std::array<double, arraySize> sinusOfNumbers{};
```

Inicjalizacja dwóch tablic, w których przechowamy wygenerowane liczby i wartości funkcji *sin()* tych

```
std::random_device randomDevice{};  
std::mt19937 randomEngine{ randomDevice() };  
std::uniform_real_distribution distribution(-100.0, 100.0);
```

Pierwsza linijka to inicjalizacja generatora liczb losowych, który służy do wygenerowania ziarna dla silnika.

Druga linijka to inicjalizacja silnika, używający do generowania liczb algorytmu Mersenne Twister i ziarna otrzymanego z *randomDevice*.

Trzecia linijka to inicjalizacja przedziału z którego generowane będą liczby.

```
std::ranges::generate(numbers, [&distribution, &randomEngine]{  
    return distribution(randomEngine);  
});
```

Korzystając z biblioteki *<algorithm>* generujemy tablicę liczb losowych. Funkcja *std::ranges::generate()* jako pierwszy argument przyjmuje tablicę do uzupełnienia, która zostanie wypełniona wartościami zwróconymi przez drugi argument, którym jest wyrażenie lambda przechwytyjące oryginalne wartości *distribution* oraz *randomEngine* i zwracające wygenerowaną liczbę losową.

```
std::ranges::generate(sinusOfNumbers, [i = 0, &numbers]() mutable {  
    return std::sin(numbers[i++]);  
});
```

Na podobnej zasadzie jak w poprzednim kroku generujemy tablicę wartości funkcji *sin()* wygenerowanych liczb. Słowo kluczowe *mutable* pozwala na modyfikację przechwyconych przez wyrażenie lambda zmiennych, gdyż te domyślnie są stałymi.

```
std::fstream fileStream{ "file.txt", std::ios::out };
```

Otwiera strumień plikowy do pliku o nazwie *file.txt*. *std::ios::out* określa, że program będzie zapisywał dane do pliku.

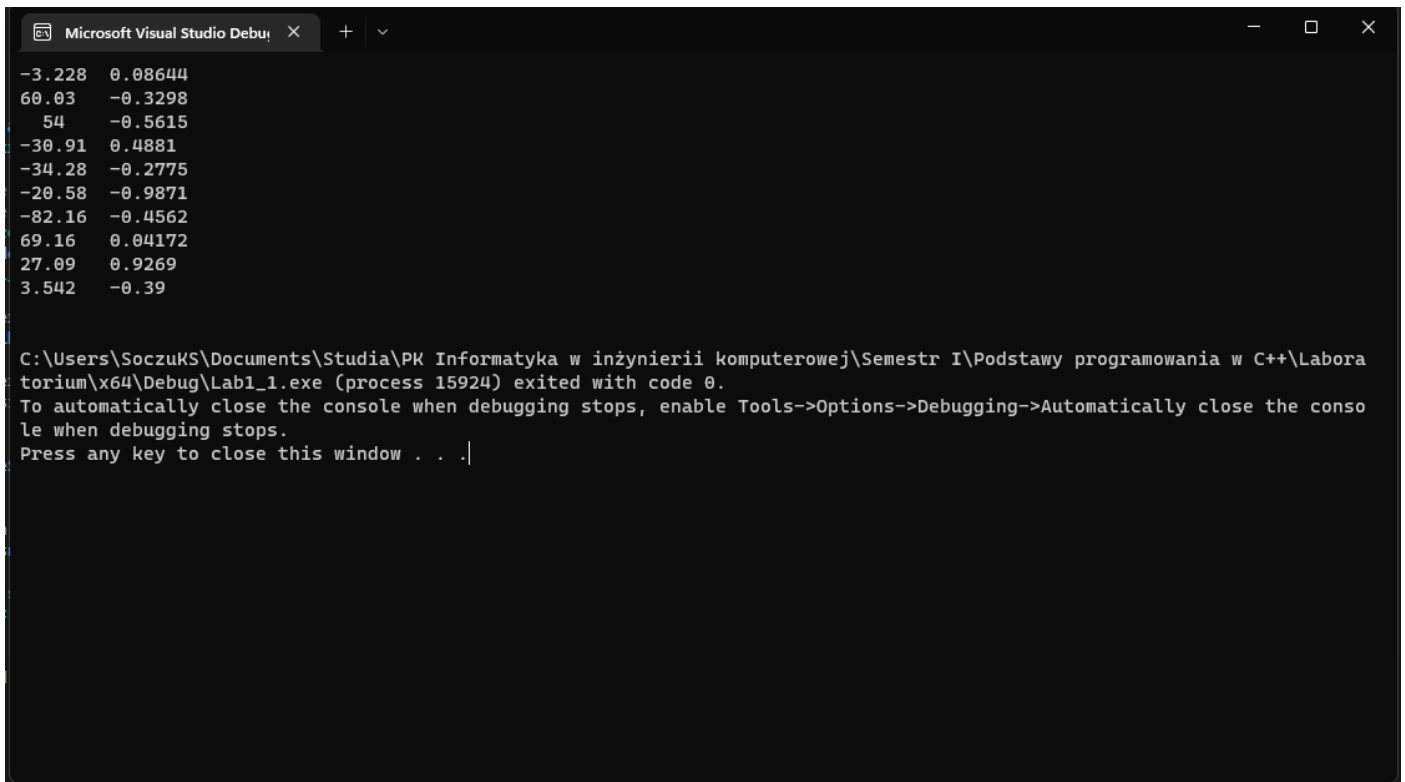
```
for (size_t i{0}; i < arraySize; ++i) {  
    const auto& n = numbers[i];  
    const auto& sn = sinOfNumbers[i];  
  
    std::cout << std::setw(streamSize) << std::setprecision(streamSize) << n << '\t' << sn << '\n';  
    fileStream << std::setw(streamSize) << std::setprecision(streamSize) << n << '\t' << sn << '\n';  
}  
std::cout << std::endl;  
fileStream << std::endl;
```

Powyższy fragment kodu z wykorzystaniem pętli *for* wypisuje obok siebie, linijka pod linijką, wygenerowane liczby i wartości funkcji *sin()* tych liczb. *setw()* określa szerokość pola kolejnej wypisanej na strumień wartości, *setprecision()* określa ilość wypisanych cyfr (szczególnie przydatne w przypadku liczb zmiennoprzecinkowych). Wypisanie na strumień *std::endl* powoduje wypisanie znaku nowej linii oraz „spłukanie” bufora danych.

```
return 0;
```

Funkcja *main()* zwraca *0* co oznacza, że program zakończył się poprawnie.

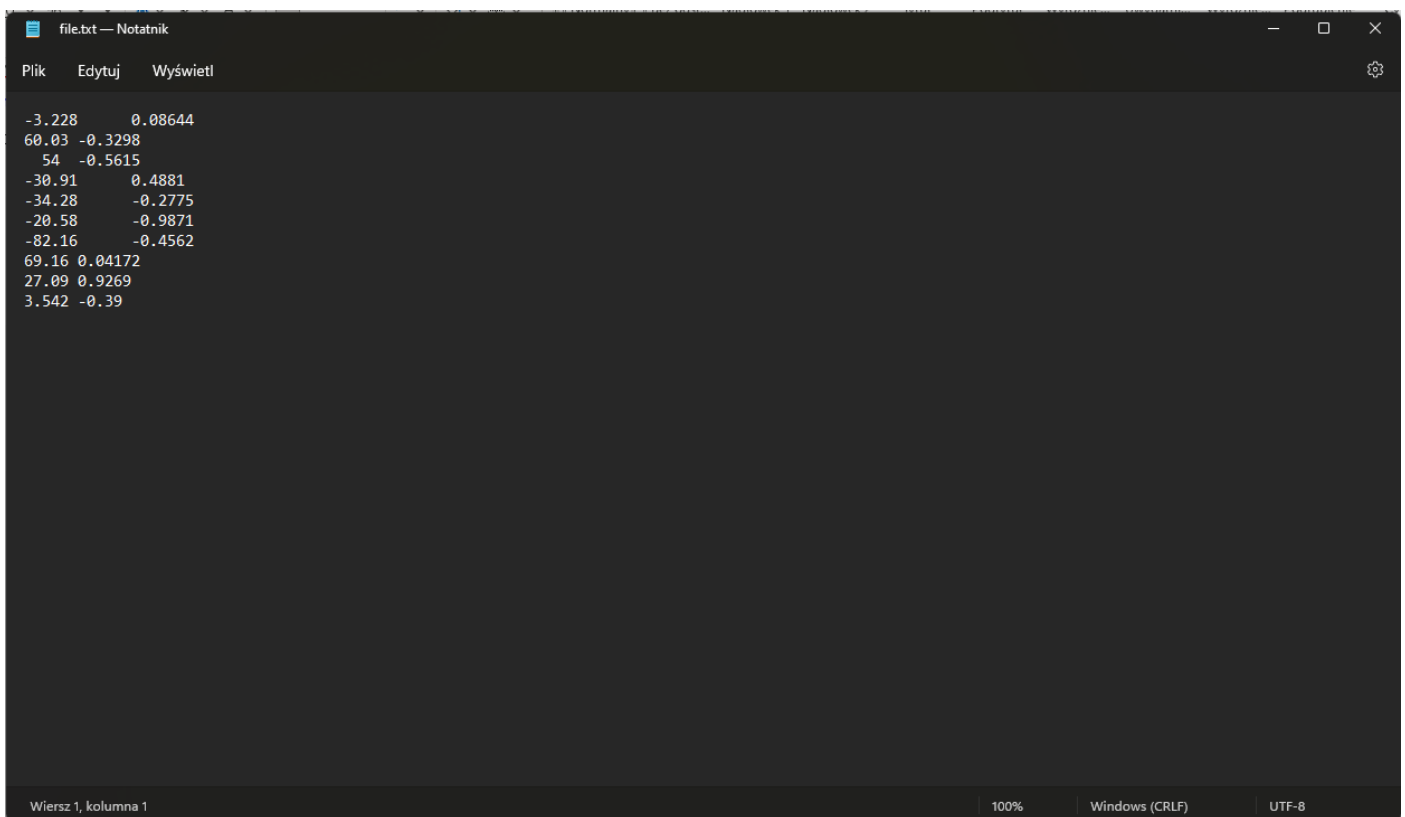
4. Wyjście programu



Microsoft Visual Studio Debug Console window showing the following output:

```
-3.228 0.08644  
60.03 -0.3298  
54 -0.5615  
-30.91 0.4881  
-34.28 -0.2775  
-20.58 -0.9871  
-82.16 -0.4562  
69.16 0.04172  
27.09 0.9269  
3.542 -0.39
```

C:\Users\SoczukS\Documents\Studia\PK Informatyka w inżynierii komputerowej\Semestr I\Podstawy programowania w C++\Laboratorium\Debug\Lab1_1.exe (process 15924) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|



Notepad window (file.txt) showing the same data as the Visual Studio console:

```
-3.228 0.08644  
60.03 -0.3298  
54 -0.5615  
-30.91 0.4881  
-34.28 -0.2775  
-20.58 -0.9871  
-82.16 -0.4562  
69.16 0.04172  
27.09 0.9269  
3.542 -0.39
```

Wiersz 1, kolumna 1 | 100% | Windows (CRLF) | UTF-8