

C++ - Podzielność liczby przez 7

© Copyright by 3bird Projects 2023, <http://edukacja.3bird.pl>

Ogólne

Sprawdzanie podzielności liczb naturalnych przez 7, stanowi w matematyce największe wyzwanie spośród wszystkich podzielności. Prawdziwą trudnością jest jednak napisanie algorytmu, który sprawdza taką podzielność bardzo dużych liczb wykraczających poza objętość poszczególnych typów danych. W takiej sytuacji należy liczbę zapisać w postaci *stringu*, potem podzielić go na 6-znakowe części i dodać je po zamianie na typ *integer*.

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 (Windows)

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    string szescOstatnichLiter;
```

```
    string liczbaJakoString = "0";
```

```
    int liczbaJakoInt, wynikModulo;
```

```
    int suma = 0;
```

```
    bool wprowadzLiczbe = true;
```

```
    cout << "=== CZY LICZBA JEST PODZIELNA PRZEZ 7? ===" << endl;
```

```
    cout << "Podaj DUŻĄ liczbę naturalną: ";
```

```
    cin >> liczbaJakoString;
```

```
    if (liczbaJakoString.length() > 6) {
```

```
        while (liczbaJakoString.length() > 6) {
```

```
            szescOstatnichLiter = liczbaJakoString.substr(liczbaJakoString.length() - 6);
```

```
            liczbaJakoInt = stoi(szescOstatnichLiter);
```

```
            suma = suma + liczbaJakoInt;
```

```
            liczbaJakoString = liczbaJakoString.erase(liczbaJakoString.size() - 6);
```

```
        }
```

```
        suma = suma + stoi(liczbaJakoString);
```

```
    }
```

```
    else {
```

```
        suma = stoi(liczbaJakoString);
```

```
    }
```

```
    wynikModulo = suma % 7;
```

```
    cout << "Sprawdzamy resztę z dzielenia przez 7: " << suma << " % 7
```

```
    = " << wynikModulo << endl;
```

```
    if (wynikModulo == 0) {
```

```

        cout << "Liczba JEST podzielna przez 7." << endl;
    }
    else {
        cout << "Liczba NIE jest podzielna przez 7." << endl;
    }

    cout << "Naciśnij ENTER, aby zakończyć..." << endl;
    system("pause > nul");
    return 0;
}

```

Kod - wersja rozbudowana (Windows)

```

#include <iostream> // Potrzebne do cout, cin, itp.
#include <windows.h> // Potrzebne do aktywacji kolorów w konsoli

using namespace std;

// ===== CZY WPROWADZONO LICZBĘ? =====
// Funkcja domyślnie zwraca prawdę, chyba że wykryje fałsz.
// Sprawdzany jest każdy znak z osobna (w pętli).
bool czyToJestLiczba(string liczbaJakoString) {
    for (int licznik = 0; licznik < liczbaJakoString.length(); licznik++) {
        if (isdigit(liczbaJakoString[licznik]) == false) {
            return false;
        }
    }
    return true;
}

// ===== MAIN =====
int main() {
    // ===== KOLOROWANIE NAPISÓW =====
    // Tworzymy tzw. uchwyt do tego, co będzie pojawiać się na konsoli (do bufora konsoli):
    HANDLE konsola = GetStdHandle(STD_OUTPUT_HANDLE);
    // Aktywujemy virtualny terminal i to, co będzie się na nim pojawiać:
    #ifndef ENABLE_VIRTUAL_TERMINAL_PROCESSING // Jeśli nie jest zdefiniowany, to:
    #define ENABLE_VIRTUAL_TERMINAL_PROCESSING 0x0004
    #endif
    // Aby powyższe działało, musimy aktywować i to, co poniżej:
    #ifndef ENABLE_PROCESSED_OUTPUT // Jeśli nie jest zdefiniowany, to:
    #define ENABLE_PROCESSED_OUTPUT 0x0001
    #endif

    // Wartość trybu (intup lub output). Słowo "dw" to skrót od "Display Window",
    // jest to jednak nazwa zmiennej, i może być inna:

```

```

DWORD dwMode = 0;
dwMode |= ENABLE_PROCESSED_OUTPUT | ENABLE_VIRTUAL_TERMINAL_PROCESSING;
SetConsoleMode(konsola, dwMode);
// ===== KOLOROWANIE NAPISÓW - KONIEC =====

string szescOstatnichLiter;
string liczbaJakoString = "0";
int liczbaJakoInt, wynikModulo;
int suma = 0;
bool wprowadzLiczbe = true; // W C++ zmienne logiczne nie mają domyślnej wartości

cout << "\n\n\033[1;34;40m=== CZY LICZBA JEST PODZIELNA PRZEZ 7? ===\033[0m" <<
endl;
cout << "\033[1;30;40mAlgorytm dzieli DUŻE liczby na grupy 6-cyfrowe (zaczynając od
tyłu)\ni sumuje te grupy. Następnie otrzymaną sumę dzieli modulo przez 7.\033[0m" <<
endl;

while (wprowadzLiczbe) {
    cout << "\n\nPodaj DUŻĄ liczbę naturalną: ";
    cin >> liczbaJakoString;
    if (czyToJestLiczba(liczbaJakoString) == false) {
        cout << "\n\n\033[1;31;40mTo nie jest liczba naturalna! Spróbuj jeszcze raz...\033[0m"
        << endl;
    }
    else {
        wprowadzLiczbe = false; // Jeśli wprowadzono liczbę, wyłącz pętlę WHILE
    }
}

try {
    // Jeśli użytkownik wpisał liczbę na tyle dużą, iż można ją podzielić na sześciocyfrowe
    // grupy, to:
    if (liczbaJakoString.length() > 6) {
        cout << "\n\nKolejne etapy sumowania sześciopaków: " << endl;

        while (liczbaJakoString.length() > 6) {
            // Nie można dopuścić, aby zmienna "szescOstatnichLiter" stała się pusta,
            // gdyż wtedy wystąpi błąd w konwersji na "liczbaJakoInt". Dlatego powyżej
            // musi być "> 6" (aby zawsze została jakaś reszta), a nie na przykład "> 5".
            // Poniżej, kopiujemy do zmiennej 6 ostatnich znaków z podanego przez użytkownika
            // stringa.
            szescOstatnichLiter = liczbaJakoString.substr(liczbaJakoString.length() - 6);
            liczbaJakoInt = stoi(szescOstatnichLiter); // Zamiana tych sześciu ostatnich znaków
            // na liczbę
            suma = suma + liczbaJakoInt; // Za każdym przejściem pętli kolejne sześciopaki
            // dodawane są do sumy.
            // Wypisujemy (diagnostycznie) aktualny stan sumy po każdym prześciu pętli:
            cout << suma << endl;
        }
    }
}

```

```

// Po każdym przejściu pętli usuwamy ze stringa 6 ostatnich znaków (skracamy go):
liczbajakoString = liczbajakoString.erase(liczbajakoString.size() - 6);
}
// Do wszystkich sum sześciopaków dodajemy na końcu pozostałą resztę:
suma = suma + stoi(liczbajakoString);
cout << "Reszta z podziału na 6-cyfrowe grupy: " << liczbajakoString << "." << endl;
cout << "Suma wszystkich sześciopaków (grup liczbowych) to: \033[1;33;40m" <<
suma << "\033[0m." << endl;
}
else {
// Jeśli użytkownik podał małą liczbę, po prostu przypisujemy ją do sumy i sprawdzamy
// dzielenie modulo:
suma = stoi(liczbajakoString);
}

wynikModulo = suma % 7;
cout << "\n\nSprawdzamy resztę z dzielenia przez 7: \033[1;33;40m" << suma << " % 7
= " << wynikModulo << "\033[0m." << endl;

if (wynikModulo == 0) {
cout << "\n\033[1;32;40mLiczba JEST podzielna przez 7.\033[0m\n\n" << endl;
}
else {
cout << "\n\033[1;31;40mLiczba NIE jest podzielna przez 7.\033[0m\n\n" << endl;
}
}
// Ten catch() jest tutaj "na wszelki wypadek", gdyż funkcja isdigit() już wymusiła poprawnie
// wprowadzone liczby:
catch (invalid_argument& it_is_not_number) {
cout << "\033[H \033[2J \033[3J" << endl; // Czyścimy ekran w przypadku wystąpienia
// błędu
cout << "\n\n\033[0;37;41mBŁĄD:\033[0m" << endl;
cout << "\033[1;31;40mTo, co wpisał użytkownik, nie jest liczbą naturalną!\033[0m\n\n"
<< endl;
// cout << "Błąd funkcji " << it_is_not_number.what() << endl; // Tylko w celach
// diagnostycznych
}

cout << "Naciśnij ENTER, aby zakończyć..." << endl;
cin.ignore(256, '\n'); // Usuwa z bufora znak \n (w ilości 256) pozostawiony przez
// wcześniejsze "cin" ("cin" domyślnie zostawia taki znak).
cin.get();
// Zamiast cin.get można by użyć system("pause > nul"), ale jest to mniej uniwersalne
// (działa tylko w Windows)
return 0;
}

```