

C++ - Silnia

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

Ogólne

W wersji podstawowej zadania, wypisywany jest wynik silni bez kontroli danych wejściowych. W wersji rozbudowanej mamy kontrolę błędów oraz formatowanie tekstu.

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;

int main() {
    int liczba;
    cout << "\n\n===== OBLICZANIE SILNI =====\n\n";
    cout << "Podaj liczbę naturalną: ";
    cin >> liczba;
    if (liczba < 1) {
        cout << "\n\nBŁĄD: Musisz podać liczbę naturalną większą od 0." << endl;
    }
    else {
        int licznik = 1;
        int silniaWynik = 1;
        while(licznik <= liczba) {
            silniaWynik = silniaWynik * licznik;
            licznik++;
        }
        cout << "\n\n" << liczba << "! = " << silniaWynik << endl;
    }
    cout << "\n\nNaciśnij ENTER, aby zakończyć..." << endl;
    system("pause > nul");
}
```

Kod - wersja rozbudowana

```
#include <iostream>
#include <string> // Potrzebne do stoi()
#include <windows.h> // Potrzebne do Sleep() i do kolorowania czcionki
using namespace std;

int main() {
    // Jeśli Twój Windows obsługuje kolorowe znaki ANSI w „Wierszu poleceń”, możesz poniższy
    // blok kodu pominąć:
```

```

// ===== KOLOROWE ZNAKI ANSI =====
// Na początek naprawiamy Windowsa, aby kolorował czcionki w konsoli.
// 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 (input 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);
// ===== KONIEC BLOKU =====

string liczbaString;
int liczbaInteger;
const int gornyZakres = 33;
cout << "\n\n\033[1;36;40m===== OBLICZANIE SILNI =====\033[0m\n\n";
cout << "Podaj liczbę naturalną: ";
getline(cin,liczbaString);
try {
    int liczbaInteger = stoi(liczbaString);
    if ((liczbaInteger < 1) || (liczbaInteger > gornyZakres)) {
        cout << "\n\n\033[0;37;41mBŁĄD:\033[0m Musisz podać liczbę naturalną w zakresie
        \033[1;36;40m0\033[0m - \033[1;36;40m" << gornyZakres << "\033[0m.\a" << endl;
    }
    else {
        int licznik = 1;
        int silniaWynik = 1;
        while(licznik <= liczbaInteger) {
            silniaWynik = silniaWynik * licznik; // np. kolejne przebiegi dla 5! = (1 * 1 = 1),
            (1 * 2 = 2), (2 * 3 = 6), (6 * 4 = 24), (24 * 5 = 120)
            licznik++;
        }
        cout << "\n\n" << liczbaInteger << "! = \033[1;32;40m" << silniaWynik << "\033[0m
        \n";
    }
}

catch (invalid_argument &e) {

```

```
//cout << e.what();  
cout << "\n\n\033[0;37;41mBŁĄD:\033[0m Nie wpisano dodatniej liczby całkowitej...  
\\a" << endl;  
cout << "\n\nZaraz nastąpi zamknięcie programu." << endl;  
Sleep(6000);  
exit(EXIT_FAILURE); // Zwraca kod błędu 1.  
}  
  
cout << "\n\nNaciśnij ENTER, aby zakończyć..." << endl;  
system("pause > nul");  
}
```

Ostatnia aktualizacja: 10 października 2023.