

# Informatyka II

## EE-DI, WEiI PRz

Instrukcja ta jest dostępna w formie źródłowej pod adresem [http://git.dms-serwis.com.pl/henrietta/inf2\\_eedi](http://git.dms-serwis.com.pl/henrietta/inf2_eedi).

Pod tym adresem będą pojawiać się również instrukcje do kolejnych laboratoriów.

## Laboratorium 3

---

Z tego laboratorium przygotowujesz sprawozdanie. Przygotowujesz je na zajęciach, a przy ich zakończeniu wysyłasz na adres podany na końcu tej instrukcji. Instrukcja dla wszystkich jest taka sama.

Rzeczy oznaczone tak, jak poniżej, dotyczą tego, co masz zawrzeć w sprawozdaniu. Na przykład:

Zapisz swoje imię, nazwisko, adres e-mail, kierunek i rok studiów oraz grupę laboratoryjną i numer albumu.

Podaj również numer zajęć laboratoryjnych (nr 3).

Mogą być to też pytania, na które w sprawozdaniu udzielisz odpowiedzi. Możesz pomagać sobie wyszukiwarką internetową, oraz zabrać głos w dyskusji, jeśli się jakaś wywiąże.

Zaliczenie najwyraźniej wypada w Dzień Dziecka (1 VI), za co szczerze przepraszam. Uratować mogą was tylko godziny rektorskie albo przejeżdżający mnie pociąg.

## Co to jest interpreter

Interpreter jest programem komputerowym, który wykonuje inny program komputerowy. Robi to w taki sposób, że czyta jego kod źródłowy - linijka po linijce - i wtedy decyduje co dopiero należy zrobić. Drugą możliwością jest kompilator - przekształca on wtedy kod programu na kod zrozumiały bezpośrednio przez procesor. Pewną analogią jest tłumaczenie:

- interpreter jest jak tłumacz symultaniczny, który tłumaczy rozmowę w jej trakcie. Jeśli jest ona monologiem, wygłoszenie jej będzie zajmowało dłużej, ale w razie problemów pozwala naprawić sytuację w trakcie
- kompilator jest jak tłumacz tłumaczący kartkę z przemówieniem. Jeśli coś pójdzie źle - nie jego sprawa. Programista będzie musiał poprawić błąd i zrobić to jeszcze raz

Pewnym rozwiązaniem (w zasadzie standardem obecnie) są kompilatory JIT

Pisanie kompilatorów jest wyższą szkołą jazdy i oczekiwanie tego od studentów na kierunku niebędącym informatyką jest czystym szaleństwem, ale prosty interpreter napisać może każdy. Spróbujemy to zrobić na tych zajęciach.

## Język programowania

Naszym językiem programowania będzie po prostu sekwencja liczb całkowitych. Stanem programu - czyli jego zmiennymi - będzie jedna zmienna symbolizująca “zapamiętaną wartość”, czyli akumulator. Taki język programowania będzie językiem regularnym oraz (w sumie jest to konsekwencja bycia jedynie językiem regularnym) nie będzie kompletny w sensie Turinga, czyli w praktyce nadawał się będzie tylko do zabawy.

Będzie on wczytywał po kolei liczby *skądś*. W naszym przypadku będzie to strumień wejściowy `std::cin`. Liczba będzie miała konkretne znaczenie, tak więc **przeczytaj specyfikację ze zrozumieniem**.

## Specyfikacja

Wartościami wejściowymi są zmienne całkowite. Błędy obsługuj zakańczając program wcześniej, podając numer liczby, z którą był problem oraz wypisując powód błędu.

Jeśli potrzebna jest kolejna liczba, a nie jest ona dostępna (bo np. skończyły się liczby z pliku), przyjmij że wczytano zero.

Gdy wykonałeś jedną komendę, następna pobrana liczba określa kolejną.

- 0 - koniec programu. Wypisz zawartość akumulatora
- 1 - pobierz kolejną liczbę. Dodaj ją do akumulatora i wynik umieść w akumulatorze
- 2 - pobierz kolejną liczbę. Pomnóż ją przez wartość akumulatora i wynik umieść w akumulatorze
- 3 - pobieraj i sumuj kolejne liczby aż nie napotkasz zera. Wynik dodaj do akumulatora

- 4 - podziel zawartość akumulatora przez kolejną pobraną liczbę. Wynik umieść w akumulatorze
- 5 - pobieraj kolejne liczby aż nie napotkasz zera. Ich średnią arytmetyczną dodaj do akumulatora. Wynik umieść w akumulatorze
- 6 - po prostu wyświetl wartość akumulatora
- 7 - pobieraj kolejne liczby aż nie napotkasz 0. Ustaw największą z nich jako wartość akumulatora
- 8 - pobieraj kolejne liczby aż nie napotkasz 0. Ustaw najmniejszą z nich jako wartość akumulatora

## Rozgrzewka i idea

Napisz w tym języku programy:

- 1) Wyświetlające wynik działania  $2+2*6$
- 2) Wyświetlające średnią arytmetyczną ciągu 2 4 5 5 3 5 2 5 1

Nie przeprowadzaj żadnych działań "na kartce", w szczególności nie zapisuj wyniku "na ostro" w programie.

Po to masz program, żeby policzył za ciebie.

## Implementacja

Kiedy już program napiszesz, spróbuj napisać jego interpreter. Będzie to program, który będzie czytał kolejne liczby - tak jak program - a następnie po prostu go wykonywał.

Załączony do archiwum z instrukcjami jest plik `program.txt`. Zawiera on program, który testuje wszystkie zachowania interpretera. Możesz uruchomić go następująco

```
g++ -o program program.cpp
program < program.txt
```

Wyświetlony powinien zostać wynik (lub błąd).

Po ukończeniu pisania daj znać prowadzącemu. Sprawdzi on Twój program.

W sprawozdaniu zamieść kompletny kod rozwiązania, oraz opracowanie programy w "liczbowym" języku programowania.

## Wyślij sprawozdanie

To już prawie koniec zajęć. Zapisz i wyślij swoje sprawozdanie.

Wyślij je na adres `sprawozdania@henrietta.com.pl`.  
W tytule umieść imię, nazwisko i numer zadania.  
Załącz sprawozdanie, lub wklej je w treść maila.

---

Copyright (c) 2017 Piotr Maślanka. Niektóre prawa zastrzeżone.

Kod źródłowy dostępny na [http://git.dms-serwis.com.pl/henrietta/inf2\\_eedi](http://git.dms-serwis.com.pl/henrietta/inf2_eedi).