Michał Bronikowski
Deklaruję zadania numer: 1,2,3,4,5,6

1. Wiadomo, że rozwiązaniem równania kwadratowego

   $$ax^2 + bx + c = 0(a \neq 0)$$

   są liczby:

   (a) $\frac{-b+\sqrt{b^2-4ac}}{2a}$

   (b) $\frac{-b-\sqrt{b^2-4ac}}{2a}$

   Pokaż na kilku przykładach, że bezpośrednie stosowanie tych wzorów może być niebezpieczne.
   Odp:
   Rozwiązanie: *zad1.rb*
   Dane: *zad1_dane.txt*

2. Użyj komputera do wyznaczania wartości numerycznych kolejnych elementów ciągu (x_n).Zdefiniowanego rekurencyjnie w następujący sposób:

   $x_0 = 1$
   $x_1 = \frac{1}{5}$
   $x_{n+2} = \frac{26}{5}x_{n+1} - x_n$

   Skomentuj otrzymane wynikii. Czy są one wiarygodne?
   Rozwiązanie:
   Zastanówmy się jaki jest wzór jawny tego ciągu.
   Niech

   $x_{n+2} = r^{n+2}$

   Wtedy:

   $$r^{n+2} = \frac{26}{5}r^{n+1} - r^n // : r^n$$

   $$r^2 = \frac{26}{5}r^1 - 1$$

   $$r^2 - \frac{26}{5}r^1 + 1 = 0$$

   Skorzystamy ze wzorów Viete'a:

$$r_1 + r_2 = \tfrac{26}{5}$$
$$r_1 \times r_2 = 1$$

Więc $r_1 = \frac{1}{5}$ a $r_2 = 5$.
Przedstawmy naszą zależność rekurencyjną dla 2 i 3 elementu naszego ciągu w postaci równania:

$x_2 = \frac{1}{25} = A \times (\frac{1}{5})^2 + B \times 5^2$

$x_3 = \frac{1}{125} = A \times (\frac{1}{5})^3 + B \times 5^3$
Wnioskujemy z tego, że:

$A = 1$

$B = 0$

Mamy więc doczynienia z ciągiem o postaci:

$x_n = (\frac{1}{5})^n$ Można zauważyć, że ciąg ten jest malejący. Przejdźmy do wyznaczenia wartości tego ciągu przy użyciu komputera.
Plik:*zad2.rb*
Po wypisaniu 30 pierwszych elementów tego ciągu mogę stwierdzić, że otrzymane przeze mnie wynikii nie są wiarygodne, ponieważ przekazywane na wyjście elementy nie są w porządku malejącym, a wręcz w pewnym momencie zaczynają rosnąć, a wykazałem, że ciąg powinien byc malejący.

3. Wykorzystując własności szeregów naprzemiennych, ustal ilu teoretycznie wyrazów szeregu:

$$\pi = 4\sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1}$$

Należy użyć do obliczenia wartości π z błędem mniejszym niż 10^{-7}
Z kryterium **Leibniza**
Jeśli ciąg a_n jest malejący i zbieżny do zera to szereg $\sum_{n=1}^{\infty}(-1)^n a_n$ jest malejący.Nasz ciąg a_n jest postaci:

$$a_k = \frac{1}{2k+1}$$

(a) Czy ciąg jest zbieżny

$$\lim_{k\to\infty} \frac{1}{2k+1} = 0$$

Ciąg jest więc zbieżny.

(b) Czy ciąg jest malejący?

$$\frac{1}{2k+1} > \frac{1}{2k+2}$$

Ciąg jest malejący.

Skoro wiem, że szereg jest zbieżny skorzystam z własności:

$$|S_{-}S_k| <= a_{k+1}$$

Cemu tak jest?
Weźmy sobie szereg naprzemienny o wyrazach kolejno równych:

a1 , a2, a3, a4, a5, a6, a7, a8, a9, ...
Ustalmy sobie S4 = a1 + a2 +a3 +a4, S - suma całości. Wiemy, że suma od a5 do an jest albo większa albo mniejsza od zera, ponieważ wyrazy możemy pogrupować w pary (a5,a6);(a7,a8) itd. Każda z tych par w zależności od wartości a5 jest albo ujemna, albo dodatnia ciąg

$$a_k = \frac{1}{2k+1}$$

jest malejący zaczynamy od liczby ujemnej więc potem dodajemy do niej dodatnią ale w module od niej mniejszą itd. Każda z tych par jest ujemna. Możemy teraz te wyrazy pogrupować tak że zostawimy sobie a5 i resztę weźmiemy w dwójki, które w zależności od porzednich np. jak były ujemne staną się dodatnie. Wiemy natomiast, że całość ma być ujemna, więc jak do ujemnej dodamy sumę tych par (> 0), to otrzymamy liczbę ujemną. Z tego wynika, że:

$$|S_-S_k| <= a_{k+1}$$

Zależy nam na tym, aby błąd był mniejszy od 10^{-7} więc:

$$|S - S_k| <= a_{k+1} < 10^{-7}$$

Dalej:

$$\frac{1}{2(k+1)+1} < 10^{-7} // \times (2k+1)$$
$$1 < 10^{-7} \times (2k+3)$$
$$k = 5000000 - 3$$
$$k = 4999997 + 1 // + 1 bomabycwikszeod1anierwne$$
$$k = 4999998$$

Teraz wykonałem odpowiedni eksperyment z wykorzystaniem komputera.
Źródło: *zad3.rb* , *testzad3.rb*
Wynika z niego, że dla k = 4999998 program nie wyznacza π z błędem mniejszym niż 10^{-7}. Bład jest mniejszy od 10^{-7} dla k = 5000000.

4. Wykorzystując własności szeregów naprzemiennych, sprawdź,że do obliczenia $\ln 2$ z błedem mniejszym niż $\frac{1}{2} \times 10^{-6}$ trzeba użyc ok. 2 milionów wyrazów szeregu:

$$\ln x = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{(x-1)^k}{k}$$

Z kryterium **Leibniza**
Jeśli ciąg a_n jest malejący i zbieżny do zera to szereg $\sum_{n=1}^{\infty} (-1)^n a_n$ jest malejący.Nasz ciąg a_n i dla x = 2 jest postaci:

$$a_k = \frac{1}{k}$$

(a) Czy ciąg jest zbieżny

$$\lim_{k \to \infty} \frac{1}{k} = 0$$

Ciąg jest więc zbieżny.

(b) Czy ciąg jest malejący?

$$\frac{1}{k} < \frac{1}{k+1}$$

Ciąg jest malejący.

Skoro wiem, że szereg jest zbieżny skorzystam z własności:

$$|S - S_k| <= a_{k+1}$$

Zależy nam na tym, aby błąd był mniejszy od $\frac{1}{2} \times 10^{-6}$ więc:

$$|S - S_k| <= a_{k+1} < \frac{1}{2} \times 10^{-6}$$

Dalej

$\frac{1}{k+1} < \frac{1}{2} \times 10^{-6} // \times 2$

$\frac{2}{k+1} < 10^{-6}$

$k + 1 = 2 \times 10^6$

$k = 2 \times 10^6 - 1$

Dalej wykaż, że wykorzystanie prostego związku

$$\ln 2 = \ln[e \times \frac{2}{e}]$$

może znacznie przyśpieszyć obliczenia.
Odp:

$$\ln[e \times \frac{2}{e}] = \ln e + \ln \frac{2}{e} = 1 + \ln \frac{2}{e}$$

Sprawdźmy teraz ilu elementów ciągu musimy użyć aby obliczyć $\ln \frac{2}{e}$ z błędem mniejszym niż $\frac{1}{2} \times 10^{-6}$. Podstawmy do naszego szeregu za x = $\frac{2}{e}$. Otrzymujemy:

$$\ln x = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{(\frac{2}{e} - 1)^k}{k}$$

Szereg ten nie jest naprzemienny, co trochę komplikuje całą sprawę.
Zaokrąglijmy więc licznik, który jest równy ok.-0.264241 do wartości równej $\frac{-1}{5}$ .Więc

$$\ln x = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{(\frac{-1}{5}^k)}{k}$$

Dalej Można zauważyć, że są to wartości asymptonicznie szybko malejące. Ponadto wszystkie ¡ 0. I już różnica $S_{11} - S_{10}$ równa jest w module $|3.9866828560608525e - 08|$. Wiedząc, że wartości będą coraz to mniejsze, a już mamy doczynienie z liczbą praktycznie niezauważalna”. Twierdzę, że do optymalnego obliczenia (z dość małym błędem) sumy tego szeregu będziemy potrzebować nie więcej niż 30 jego wyrazów. (Dalsze iteracje będą zbyt mało znaczące).

5. Wykorzystując pomysł z poprzedniego zadania zaproponuj szybki algorytm do wyznaczania ln bardzo dużych liczb.

```
def calc(x,k)
    sum = 0
    for i in 1..k
        sum+=(((-1) ** (i-1) ) * (((x-1) ** i) / i))
    end
    sum
end

```

```ruby
def ln x
    if  x > 5
      temp = x / Math::E
     return 1 + 1 + ln(temp/Math::E)
    else
     return 1 + calc((x / Math::E),30)
    end
end
```

Listing 1: Algorytm do wyznacznia logarytmu naturalnego

Opis:

Na początku sprawdzam czy **x** jest większy od 5, aby podczas dzielenia przez **e** wynik nie był większy od 2. Następnie wyciągam z tej liczby **e** i do wyniku dodaje **1** i rekurencyjnie wywołuję funkcję **ln** z tym co mi zostało po wyciągnięciu e. Jeżeli argument funkcji **ln** jest mniejszy równy 5 to postępuje zgodnie z pomysłem z poprzedniego zadania.

Test: *testzad5.rb*

6. W języku **PWO ++** funkcja **ATAN** oblicza arc tgz bardzą dużą dokłądnością, ale tylko wtedy gdy $|x| \leqslant 1$. Zaproponuj algorytm wyznaczający w jęzuku **PWO ++** arc tg z dużą dokładnością dla $|x| > 1$.

```ruby
def ATan(x)
    return Math.atan(x) if x.abs <= 1
    print "Wywolano funkcje ATan z |x| > 1"
end

def FATan(x)
    if x < 0
        a = - (Math::PI / 2.0)
     return a - ATan(1.0/x)
    elsif x == 0
     return ATan(x)
    else
        a = (Math::PI / 2.0)
     return a - ATan(1.0/x)
    end
end
```

Listing 2: Kod w Ruby

Korzystam z faktu, że gdy $x >= 0$ to:

$$\operatorname{arc\,tg}(x) + \operatorname{arc\,tg}(\frac{x}{2}) = \frac{\pi}{2}$$

A gdy $x < 0$ to:

$$\operatorname{arc\,tg}(x) + \operatorname{arc\,tg}(\frac{x}{2}) = -\frac{\pi}{2}$$