**Dokumentacja projektu**

PRI - zadanie laboratoryjne 1

**Autor:**

Kamil Lipski

Warszawa, 2016

1. **Cele**

Program oblicza wartość podanej całki: korzystając z przedstawionego rozwinięcia: 

dla różnych wartości x z przedziału <0,1> z dokładnością

ε.

1. **Opis działania programu**

*Zmienne:* x - wartość podawana przez użytkownika, epsilon - dokładność obliczeń (podawana przez użytkownika), S1, S2, S3 - zmienne służące do prze- prowadzania częściowych obliczeń, W - zmienna służąca do porównania uzyskanego wyniku z epsilon, cc - część całkowita wyniku, cu - część ułamkowa wyniku, k1, k2 - wartości liczbowe mnożone przez silnię z kolejnych mianowników, potega1, potega2 - kolejne potęgi, s, n, wynik1, wynik2 - zmienne służące do liczenia silni, warunki - zmienna służąca do wykonywania pętli while

*Użyte funkcje:*

* pow(a,b) - funkcja zwraca wartość działania ab,
* fabs(a) - funkcja zwraca wartość bezwzględną   
  z wartości zmiennej a,
* a=modf(L, &b) - w tym zapisie funkcja modf przypisuje zmiennej a część ułamkową wartości zmiennej L,   
  a zmiennej b - część całkowitą wartości zmiennej L.

Po podaniu wartości zmiennych x i epsilon przez użytkownika, program pobiera je i sprawdza ich poprawność (w przypadku x spoza przedziału <0,1>, epsilon>0.1, epsilon≤0, epsilon>x, czy podaniu przez użytkownika wartości innych niż liczbowe, program wypisuje stosowny komunikat - zagadnienie opisałem szerzej w założeniach). Program wykonuje stosowne podstawienia (m.in. S1=x), liczy pierwszą silnię i wy- konuje działanie: .



Program wykonuje kolejne podstawienia i oblicza wartość działania: . Następnie sprawdza, czy |S3-S1| <epsilon. Jeśli nie, wykonuje dalsze działania. Jeśli tak,   
to zaokrągla wynik (S3) do podanej dokładności i wy- świetla go użytkownikowi.



**3. Założenia**

* użytkownik programu podaje x z przedziału <0,1> -   
  w przeciwnym wypadku program wypisuje błąd,
* użytkownik programu podaje dokładność ε, która jest   
  ≤ 0.1 (narzucone przeze mnie ograniczenie górne). ε nie powinien być też większy od x - wykonywanie obliczeń byłoby bezcelowe. W powyższych przypadkach program wypisuje komunikat: „Za duzy epsilon”. Jeśli użytkownik poda epsilon ≤0, program wypisze komunikat: „Niewlasciwy epsilon!” (nie można liczyć   
  z dokładnością równą 0 lub z ujemną dokładnością). Dodatkowo zakładam, że ε podany przez użytkownika będzie jedną z potęg liczby 10, a maksymalna dokładność oczekiwana przez użytkownika będzie wynosiła 6 miejsc po przecinku (ograniczenia zastosowanego typu zmiennych nie pozwalają   
  na większą),
* gdy użytkownik poda wartości inne niż liczbowe, program wypisze błąd.
* podanie dokładności ε skutkuje tym, że jeśli w toku działania programu uzyskiwane wyniki będą różnić się   
  o wartość mniejszą od ε, to program przerwie działanie (dalsze obliczenia miałyby zaniedbywalny wpływ   
  na pożądany wynik) i zaokrągli do podanej dokładności to, co do tej pory uzyskał.

**4. Przykładowe testy**

* x=a, epsilon=a - komunikat: „Niepoprawne dane”,
* x=2, epsilon=0.01 - komunikat: „x spoza zbioru”,
* x=-1, epsilon=0.01 - komunikat: „x spoza zbioru”,
* x=0.1, epsilon=-0.1 - komunikat: „Niewlasciwy epsilon!”,
* x=0.01, epsilon=0.1 - komunikat: „Za duzy epsilon”,
* x=1, epsilon=0.2 - komunikat: „Za duzy epsilon”,
* x=0.5, epsilon=0.01 - wynik: 0.46 (wynik   
  w WolframAlpha: 0.461281) - obliczenia poprawne,
* x=0.6, epsilon=0.0001 - wynik: 0.5352 (wynik   
  w WolframAlpha: 0.535154) - obliczenia poprawne.

**5. Schemat blokowy**

