**Министерство образования и наук РФ**

**Федеральное государственное образовательное учреждение**

**Высшего образования**

**Рязанский государственный радиотехнический университет**

**Кафедра ЭВМ**

Отчет по лабораторной работе №4:

“Итерационные циклы”

Вариант 12

Выполнила: ст. гр 640

Тограева К. О.

Проверила:

Доц. к.т.н. Елесина С.И.

Рязань 2017

**Цель работы:**  
 получение навыков написания программ с итерационными циклами на языке Си.

**Вариант №11:  
Часть 1.**

Реализация N - арных операций

Задания

– найти обобщенную формулу (или формулы) вычисления;

– разработать алгоритм вычисления значения переменной S (см. задания) и построить на экране таблицу ее значений в виде:

|  |  |
| --- | --- |
| i | S[i] |
| 1 |  |
| 2 |  |
| 3 |  |

Замечание: S[i]–все промежуточные значения, полученные в процессе вычислений.

S = 1/(1\*2\*3) + 1/(2\*3\*4) + 1/(3\*4\*5) + 1/(4\*5\*6)

**Часть 2.**

Уточнение корней уравнений методом половинного деления

Для заданного уравнения и заданного интервала, содержащего один корень, используя метод половинного деления, уточнить корень нелинейного уравнения с точностью до

ε = 10-5.

На экране монитора вывести границы интервала, получаемые при его делении пополам (для всех вариантов число таких интервалов не превышает 18).

После выхода из итерационного цикла корень считать равным середине интервала и провести проверку путем подстановки в исходное уравнение, чтобы убедиться в правильности нахождения корня.

Уравнение:

X + cos(x0.52 + 2) = 0

Область [0.5; 1]

**Программирование**

**#include <stdio.h>  
#include <math.h>  
#include <stdlib.h>  
  
double calculate\_uravnie(double x);  
  
int main() {  
 double sum = 0;  
 printf("╔══════╤════════╗\n");  
 printf("║ i │ S[i] ║\n");  
 printf("╟──────┼────────╢\n");  
 for (int i = 1; i < 4; ++i) {  
 sum += (double) 1 / (i \* (i + 1) \* (i + 2));  
 printf("║ %i │ %.3lf ║\n", i, sum);  
 if (i != 3) {  
 printf("╟──────┼────────╢\n");  
 }  
 }  
 printf("╚══════╧════════╝\n\n\n");  
  
 printf("SECOND PART\n");  
  
 double l\_board = 0.5;  
 double r\_board = 1;  
 double seredina;  
 double znac\_l, znac\_r, znac\_s;  
 double e = 10e-5;  
  
 while (fabs(l\_board-r\_board) > e){  
 znac\_l = calculate\_uravnie(l\_board);  
 znac\_r = calculate\_uravnie(r\_board);  
 seredina = (l\_board + r\_board) / 2;  
 znac\_s = calculate\_uravnie(seredina);  
  
 printf("%lf % lf % lf\n", l\_board, seredina, r\_board);  
 printf("Левая граница, середина, правая граница\n\n");  
  
 if (znac\_r > 0 & znac\_s > 0) {  
 r\_board = seredina;  
 } else {  
 l\_board = seredina;  
 }  
 }  
 // проверка  
 printf("%lf", calculate\_uravnie(seredina));// значение должно быть около 0  
}  
  
double calculate\_uravnie(double x) {  
 return x + cos(pow(x, 0.52) + 2);  
}**

Тестовый пример:

Пример

╔══════╤════════╗

║ I │ S[i] ║

╟──────┼────────╢

║ 1 │ 0.167 ║

╟──────┼────────╢

║ 2 │ 0.208 ║

╟──────┼────────╢

║ 3 │ 0.225 ║

╚══════╧════════╝

SECOND PART

0.500000 0.750000 1.000000

Левая граница, середина, правая граница

0.750000 0.875000 1.000000

Левая граница, середина, правая граница

0.875000 0.937500 1.000000

Левая граница, середина, правая граница

0.937500 0.968750 1.000000

Левая граница, середина, правая граница

0.968750 0.984375 1.000000

Левая граница, середина, правая граница

0.984375 0.992188 1.000000

Левая граница, середина, правая граница

0.984375 0.988281 0.992188

Левая граница, середина, правая граница

0.988281 0.990234 0.992188

Левая граница, середина, правая граница

0.988281 0.989258 0.990234

Левая граница, середина, правая граница

0.988281 0.988770 0.989258

Левая граница, середина, правая граница

0.988770 0.989014 0.989258

Левая граница, середина, правая граница

0.989014 0.989136 0.989258

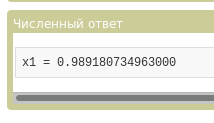
Левая граница, середина, правая граница

0.989136 0.989197 0.989258

Левая граница, середина, правая граница

0.000015

Реальный корень (для проверки)



Вывод:

Мы научились работать с итерационными циклами.