

1. **Цель работы**

Изучение способов организации циклических вычислений с известными и неизвестным числом итераций.

1. **Постановка задачи**

Вариант 1: Вычислить сумму первых N элементов ряда: .

1. **Формализация задачи**

Заданное количество элементов ряда не может быть отрицательным: n<0. При нулевом количестве элементов (n==0) подсчёт не производится, а сумма сразу выводится как 0. Числитель каждого элемента совпадает с его порядковым номером, а знаменатель в два раза больше знаменателя предыдущего элемента. Знаменатель первого элемента равен единице. Знаки элементов чередуются начиная с «+».

1. **Набор тестовых примеров**

|  |  |
| --- | --- |
| Количество элементов | Вывод программы |
| -96 | The amount of elements should positive. Please try again. |
| 243436 | The amount of elements must be equal to or less than 4000. Please try again. |
| 0 | 0 |
| 4 | 0.25 |
| 6 | 0.375 |
| 7 | 0.484375 |

1. **Схема алгоритма**

Схема алгоритма программы представлена на рисунке 1.

Начало

N = 4000, a[N] = {0},

n, i = 0, sum = 0,

denominator = 1, endFlag = 1.

Ввод n

A

B

A

B

n > N

Нет

Да

Вывод «The amount of elements must be equal to or less than 4000. Please try again.»

n < 0

Нет

Да

Вывод «The amount of elements should positive. Please try again.»

n != 0

Нет

Да

a[i] = 1;

sum = a[i];

i=1;

i < n

Нет

Да

denominator=denominator\*2;

a[i]=pow((-1),i) \* ((i+1) / (denominator));

sum += a[i];

Вывод результата sum

C

D

1 - Count again.

2 - Exit.

D

C

1

2

Конец

*Рисунок 1 – Схема алгоритма программы.*

1. **Листинг программы**

#if defined(\_WIN32) || defined(\_\_WIN32\_\_) || defined(WIN32)

#define \_CRT\_SECURE\_NO\_WARNINGS

#pragma warning(disable:4996)

#pragma warning(disable:6031)

#endif

#include <stdio.h>

#include <math.h>

#define N 4000 //defining a constant for maximum amount of elements

int main() {

float a[N] = {0}; // array of elements

int n; //the amount of elements to count

int i = 0; //counter

float sum = 0; //sum of elements

float denominator = 1; //the denominator of a fraction

int endFlag = 1; //a flag that stops the application

do{ //a global cycle that prevents the program from endting without the user's desire

printf("Please input the amount of elements.\n");

scanf("%d", &n); //input of the amount of elements

if (n > N) { //checking that the number entered by the user does not exceed the maximum allowed

printf("The amount of elements must be equal to or less than %d. Please try again.\n\n", N);

endFlag = 1;

}

else{

if(n < 0) { //checking that the number entered by the user is not negative

printf("The amount of elements should positive. Please try again.\n\n");

endFlag = 1;

}

else{

if (n != 0) { //if the amount of elements is not 0, then counting

a[i] = 1; //setting the first element as 1

sum = a[i]; //setting the sum as the value of the first element

for (i = 1; i < n; i++) {

denominator = denominator \* 2; //multiplying the denominator by two

a[i] = pow((-1),i) \* ((i + 1) / (denominator));

sum += a[i]; //adding the value of the new element to the sum

}

}

printf("\tThe result is: %f\n\n", sum); //output of the result

printf("\nPlease choose from the following:\n\t1 - Count again.\n\t2 - Exit.\n"); //suggesting to user to run program again or exit.

scanf("%d", &endFlag);

switch (endFlag) {

case 1: endFlag = 1; break;

case 2: endFlag = 0; break;

default: printf("Incorrect answer. The program will be closed."); endFlag = 0; break;

}

}

}

} while (endFlag); //if flag is 0, then exiting out of cycle

return 0;

}

1. **Выводы**

В языке Си существует следующий цикл с предусловием: while.

В языке Си существует следующий цикл с постусловием: do…while.

Отличие цикла с предусловием от цикла с постусловием в том, что цикл с постусловием точно выполнится хотя бы один раз, а цикл с предусловием может не выполниться ни одного.

Графическое представление цикла с предусловием выглядит как блок условия, за которым следует блок операций.

Графическое представление цикла с постусловием выглядит как блок операций, за которым следует блок условия.

Оператор break прерывает работу цикла или условного оператора, в котором он находится.

Оператор continue позволяет сразу перейти в конец тела цикла, пропуская весь код, который находится под ним.

Вложенные циклы – это циклы, размещённые внутри другого цикла.