ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | М. А. Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

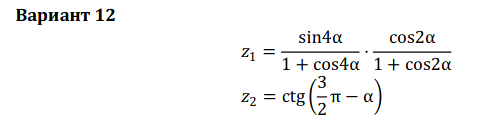
|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Вычисление математических выражений |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

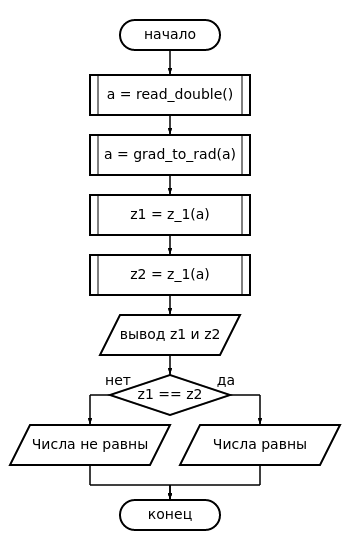
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4135К |  |  |  | Столяров Н.С. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**1.Цель работы:** Изучить вычисление математических функций в с++.  
**2.Задачи работы:** Согласно варианту 12 написать программу для расчѐта двух выражений. Предварительно подготовить тестовые примеры по обеим формулам (в excel) результат вычисления по первой формуле должен совпадать со второй. Значение параметров тригонометрических функций должны задаваться пользователем в градусах.

  
**3.Описание функций:  
1)Имя: main**   
Назначение: передача данных в функции z\_1 и z\_2 и получение значений из них и последующее их сравнение  
Входные данные: нет.   
Выходные данные: нет.  
Побочный эффект: отсутствует.   
Тестовые данные:

|  |  |  |  |
| --- | --- | --- | --- |
| Значение | 1 Функция | 2 Функция | Радианы |
| 30 | 0,577350269189626 | 0,577350269189626 | 0,523598775598299 |
| 15 | 0,267949192431123 | 0,267949192431123 | 0,261799387799149 |
| 10 | 0,176326980708465 | 0,176326980708465 | 0,174532925199433 |

Прототип: int main()  
Псевдокод:  
Функция проверки ввода данных read\_double()

Перевод градусы в радианы через функцию grad\_to\_rad()

Функция z\_1

Функция z\_2

Вывод в консоль значения z\_1

Вывод в консоль значения z\_2  
Сравнение z\_1 и z\_2  
Блок-схема:

**2)Имя: read\_double**   
Назначение: проверка переменной a на корректность ввода  
Входные данные: a(значение в градусах).   
Выходные данные: a(значение в градусах).  
Побочный эффект: отсутствует.   
Прототип: double read\_double()  
  
Псевдокод:  
Цикл продолжается до тех пор, пока пользователь не введет корректное значение

Ввод числа x

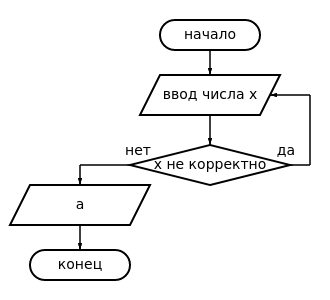
Если предыдущее извлечение оказалось неудачным, то...

Возвращаем scanf в 'обычный' режим работы(обнуление битов состояния)

Удаляем значения предыдущего ввода из входного буфера

Если всё хорошо, то возвращаем x

Возврат значения x

Блок-схема:

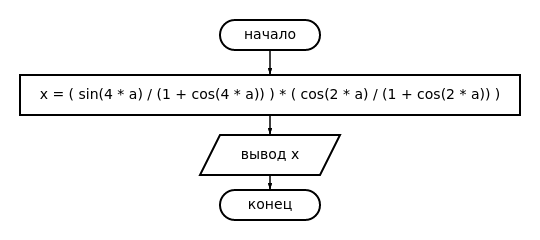
**3)Имя: z\_1**  
Назначение: высчитывание функции z1 = ( sin(4 \* a) / (1 + cos(4 \* a)) ) \* ( cos(2 \* a) / (1 + cos(2 \* a)) )  
Входные данные: a(значение в градусах).   
Выходные данные: z1(значение функции z1).  
Побочный эффект: отсутствует.   
Тестовые данные:

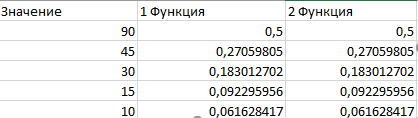
|  |  |
| --- | --- |
| Значение | 1 Функция |
| 30 | 0,577350269189626 |
| 15 | 0,267949192431123 |
| 10 | 0,176326980708465 |

Прототип: double z\_1(double a)  
  
Псевдокод:  
Высчитывания значения функции z1

Возврат значения z1

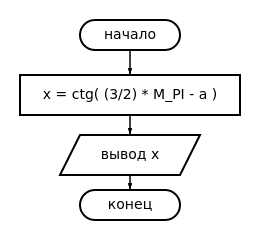
Блок-схема:

**4)Имя: z\_2**  
Назначение: высчитывание функции z2 = (sqrt(2) / 2) \* sin(a / 2)

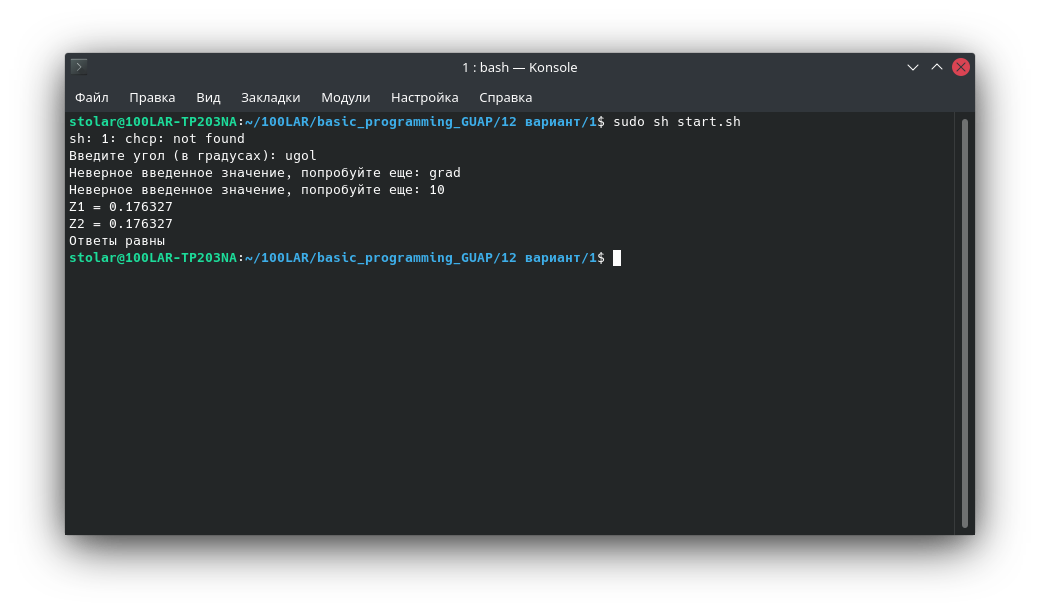
Входные данные: a(значение в градусах).   
Выходные данные: z2(значение функции z2).  
Побочный эффект: отсутствует.   
Тестовые данные:  
  
Прототип: float z\_2(double a)  
  
Псевдокод:  
Высчитывания значения функции z2

Возврат значения z2

Блок-схема:

**9.ЛИСТИНГ КОДА:**/\*   
 Напишите программу для расчѐта двух выражений. Предварительно подготовьте тестовые   
 примеры по обеим формулам (в excel или с помощью калькулятора; результат вычисления по   
 первой формуле должен совпадать со второй). Значение параметров тригонометрических функций   
 должны задаваться пользователем в градусах.   
\*/   
  
#include <stdio.h>   
#include <stdlib.h>   
#include <locale.h> // Для смены языка   
#include <math.h>   // Математические функции и константы   
  
// Функция для перевода градусов в радианы   
double grad\_to\_rad(double deg) {   
 return (deg / ( 180 / M\_PI ));   
}   
  
// Функция для рассчёта котангенса   
double ctg(double rad) {   
 return tan(M\_PI/2 - rad);   
}   
  
// Функция для проверки ввода   
double read\_double(){   
       double x;   
  
 while ( (scanf("%lf",&x) ) != 1 ) {   
     printf("Неверное введенное значение, попробуйте еще: ");   
     while(getchar() != '\n');   
 }   
 return x;   
}   
  
double z\_1(double a){   
 return ( sin(4 \* a) / (1 + cos(4 \* a)) ) \* ( cos(2 \* a) / (1 + cos(2 \* a)) );   
}   
  
double z\_2(double a){   
 return ctg( (3/2) \* M\_PI - a );   
}   
  
  
// основной код   
int main() {   
 // смена кодировки   
 system("chcp 65001");   
  
 // ввод угла в градусах   
 printf("Введите угол (в градусах): ");   
 double a = read\_double();   
  
 // перевод градусов в радианы   
 a = grad\_to\_rad(a);   
  
 // первое выражение   
 double z1 = z\_1(a);   
 printf("Z1 = %f\n", z1);   
  
 // второе выражение   
 double z2 = z\_1(a);   
 printf("Z2 = %f\n", z2);   
  
 if (z1 == z1) {   
   printf("Ответы равны\n");   
 } else {   
   printf("Ответы не равны\n");   
 }   
  
 return 0;   
}

**10.Пример выполнения программы:**



Видно, что результаты расчётов совпадают с тестовыми данными.

**11.Анализ результатов и выводы:**

В ходе этой лабораторной работы мы научились запрашивать у пользователя данные для ввода и работать с ними.

Из недостатков можно отметить, что программа пропускает на ввод данные вида «Число пробел символ» и наоборот и работает с ними.

Из плюсов можно отметить два в i++.