МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт комплексной безопасности и специального приборостроения
Кафедра КБ-2 «Прикладные информационные технологии»

А.А. МЕРСОВ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКОЙ РАБОТЫ №2

“ Арифметические операции и математические функции языка С ”

по дисциплине: **«Языки программирования»**
(наименование дисциплины)

Москва – 2021

УДК
ББК

Печатается по решению редакционно-издательского совета «МИРЭА – Российский технологический университет»

Мерсов А.А.

Методические указания по выполнению практической работы № 2 по языкам программирования / А.А. Мерсов– М.: МИРЭА – Российский технологический университет, 2021.

Методические указания предназначены для выполнения практической работы по дисциплине «Языки программирования» и содержит перечень вариантов практической работы, а также краткое изложение теоретического материала в форме пояснений к заданию на работу. Для студентов, обучающихся по направлениям 09.03.02, 10.03.01, 10.05.02, 10.05.03, 10.05.04.

Материалы рассмотрены на заседании учебно-методической комиссии КБ-2 Протокол №1 от «28» августа 2021 г. и одобрены на заседании кафедры КБ-2.

зав. кафедрой КБ-2

к.т.н. _____ / О.В.Трубиенко /

УДК ББК

Содержание

Общие указания к выполнению практической работы

Практические работы выполняются с использованием персональных компьютеров. Указания по технике безопасности совпадают с требованиями, предъявляемыми к пользователю ЭВМ. Другие опасные факторы отсутствуют.

Цель практической работы

Цель практической работы по дисциплине «Языки программирования» состоит в закреплении и углублении знаний и навыков, полученных при изучении построения алгебраических выражений и использовании математических функций библиотеки языка С. Практическая работа предполагает выполнение задания разработке и тестированию программного обеспечения.

Основные сведения из языков программирования
Арифметические операции

Арифметические операции производятся над числами. Значения, которые участвуют в операции, называются операндами. В языке программирования C/C++ арифметические операции могут быть бинарными (производятся над двумя операндами) и унарными (выполняются над одним операндом). К бинарным операциям относят следующие:

- +

  Операция сложения возвращает сумму двух чисел:

```
int a = 10;
int b = 7;
int c = a + b; // 17
int d = 4 + b; // 11
```

- -

  Операция вычитания возвращает разность двух чисел:

```
int a = 10;
int b = 7;
int c = a - b;   // 3
int d = 41 - b; // 34
```

- *

  Операция умножения возвращает произведение двух чисел:

```
int a = 10;
int b = 7;
int c = a * b;   // 70
int d = b * 5;   // 35
```

- /

  Операция деления возвращает частное двух чисел:

```
int a = 20;
int b = 5;
int c = a / b;              // 4
double d = 22.5 / 4.5;  // 5
```

  При делении стоит быть внимательным, так как если в операции участвуют два целых числа, то результат деления будет округляться до целого числа, даже если результат присваивается переменной float или double:

```
double k = 10 / 4;   // 2
printf(“%f”, k);
```

Чтобы результат представлял число с плавающей точкой, один из операндов также должен представлять число с плавающей точкой:

```
double k = 10.0 / 4;   // 2
printf("%f", k);
```

- %

  Операция получения остатка от целочисленного деления:

```
int a = 33;
int b = 5;
int c = a % b;        // 3
int d = 22 % 4;     // 2 (22 - 4*5 = 2)
```

Также есть две унарные арифметические операции, которые производятся над одним числом: ++ (инкремент) и -- (декремент). Каждая из операций имеет две разновидности: префиксная и постфиксная:

- Префиксный инкремент.

  Увеличивает значение переменной на единицу и полученный результат используется как значение выражения ++x

```
int a = 8;
int b = ++a;
printf("%d \n", a);    // 9
printf("%d \n", b);    // 9
```

- Постфиксный инкремент.

  Увеличивает значение переменной на единицу, но значением выражения x++ будет то, которое было до увеличения на единицу

```
int a = 8;
int b = a++;
printf("%d \n", a);    // 9
printf("%d \n", b);    // 8
```

Префиксный декремент.

Уменьшает значение переменной на единицу, и полученное значение используется как значение выражения –x

```
int a = 8;
int b = --a;
printf("%d \n", a);    // 7
printf("%d \n", b);    // 7
```

- Постфиксный декремент.

Уменьшает значение переменной на единицу, но значением выражения x-- будет то, которое было до уменьшения на единицу

```
int a = 8;
int b = --a;
printf("%d \n", a);    // 7
printf("%d \n", b);    // 8
```

Арифметические операции вычисляются слева направо. Одни операции имеют больший приоритет чем другие и поэтому выполняются вначале. Операции в порядке уменьшения приоритета:

++ (инкремент), -- (декремент)

* (умножение), / (деление), % (остаток от деления)

+ (сложение), - (вычитание)

Приоритет операций следует учитывать при выполнении набора арифметических выражений:

```
int a = 8;
int b = 7;
int c = a + 5 * ++b;    // 48
printf("%d", c);
```

Хотя операции выполняются слева направо, но вначале будет выполняться операция инкремента ++b, которая увеличит значение переменной b и возвратит его в качестве результата, так как эта операция имеет больший приоритет. Затем выполняется умножение 5 * ++b, и только в последнюю очередь выполняется сложение a + 5 * ++b

Скобки позволяют переопределить порядок вычислений. Например:

```
int a = 8;
int b = 7;
int c = (a + 5) * ++b;    // 104
printf("%d", c);
```

Несмотря на то, что операция сложения имеет меньший приоритет, но вначале будет выполняться именно сложение, а не умножение, так как операция сложения заключена в скобки.

Математические функции

Стандартные математические функции находятся в файле ***math.h***.
Общий вид математических функций:
тип возвращаемого значения название функции (список аргументов над которыми необходимо выполнить определенные действия).
Пример:
Получение модуля числа

```
int x=-20;
int z;
z=abs(x);   // 20
```

abs - абсолютное значение целого числа - |***x***|
int abs(int x);
labs - абсолютне значення "длинного" целого числа - |***x***|:
long labs(long x);
fabs - абсолютное значение числа с плавающей точкой - |***x***|:
double fabs(double x);
sqrt - извлечение квадратного корня:
double sqrt(double x);
pow - возведение в степень:
double pow(double x, double y);
cos - косинус - ***cos x*** (здесь и далее ***x*** задается в радианах):
double cos(double x);
sin - синус - ***sin x***:
double sin(double x);
tan - тангенс - ***tg x***:
double tan(double x);
acos - арккосинус - ***arccos x***:
double cos(double x);
asin - арксинус - ***arcsin x***:
double sin(double x);
atan - арктангенс - ***arctg x***:
double atan(double x);
atan2 - арктангенс - ***arctg x/y***:
double atan2(double x, double y);
exp - експонента :
double exp(double x);
log - натуральный логарифм - ***ln x***:
double log(double x);
log10 - десятичный логарифм - $\boldsymbol{\log_{10} x}$:
double log10(double x);

Варианты заданий

Составьте программу, которая подсчитывает и выводит значение по формулам, которые приведены в Вашем варианте индивидуального задания. Определите области допустимых значений параметров формул и задайте произвольные значения из этих областей. Значения параметров с именами x и y должны вводиться с клавиатуры, значения остальных - задаваться как начальные значения при объявлении соответствующих переменных. Допускается (и даже желательно) упростить / разложить формулы для того, чтобы обеспечить минимизацию объема вычислений .

Вариант 00

$$t1 = \frac{1}{b^3}\left(ln\frac{y}{x} - \frac{a^2x^2}{2y^2}\right)$$

$$t2 = \frac{1}{a}tg\frac{ax}{2} + \frac{1}{a}ln\ tg\frac{ax}{2}$$

0. $$t = \frac{2\cos\left(x - \frac{\pi}{6}\right)}{0.5 + \sin^2 y}\left(1 + \frac{z^2}{3 - z^2/5}\right).$$

При x=14.26, y=-1.22, z=3.5×10^{-2} **t=0.564849.**

1. $$u = \frac{\sqrt[3]{8 + |x-y|^2 + 1}}{x^2 + y^2 + 2} - e^{|x-y|}\left(tg^2 z + 1\right)^x.$$

При x=-4.5, y=0.75×10^{-4},z=$0.845\times\mathbf{10}^2$ **u=-55.6848.**

2. $$v = \frac{1 + \sin^2(x+y)}{\left|x - \frac{2y}{1 + x^2y^2}\right|}x^{|y|} + \cos^2\left(arctg\frac{1}{z}\right).$$

При x=3.74×10^{-2}, y=-0.825, z=$0.16\times\mathbf{10}^2$, **v=1.0553.**

3. $$w = |\cos x - \cos y|^{(1+2\sin^2 y)}\left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right).$$

При x=0.4×10^4 ,y=-0.875, z=$-0.475\times\mathbf{10}^{-3}$ **w=1.9873.**

4. $$\alpha = \ln\left(y^{-\sqrt{|x|}}\right)\left(x - \frac{y}{2}\right) + \sin^2 arctg(z).$$

При x=-15.246, y=4.642×10^{-2}, z=$20.001\times\mathbf{10}^2$ α **=-182.036.**

5. $\beta = \sqrt{10\left(\sqrt[3]{x} + x^{y+2}\right)}\left(\arcsin^2 z - |x - y|\right)$

При x=16.55×10^{-3}, y=-2.75, z=0.15 β**=-40.630.**

6. $y = 5arctg(x) - \frac{1}{4}\arccos(x)\frac{x + 3|x - y| + x^2}{|x - y|z + x^2}.$

При x=0.1722, y=6.33, z=3.25×10^{-4} γ **=-205.305.**

7. $\varphi = \frac{e^{|x-y|}|x - y|^{x+y}}{arctg(x) + arctg(z)} + \sqrt[3]{x^6 + \ln^2 y}.$

При x=-2.235×10^{-2}, y=2.23, z=15.221 φ **=39.374.**

8. $\psi = \left|x^{\frac{y}{x}} - \sqrt[3]{\left|\frac{y}{x}\right|}\right| + (y - x)\frac{\cos y - \frac{z}{(y - x)}}{1 + (y - x)^2}.$

При x=1.825×10^{2}, y=18.225, z=-3.298×10^{-2} ψ **=1.2131.**

Пример выполнения (вариант 00)

Разработка алгоритма решения

Основной алгоритм

Алгоритм решения задачи - линейный и состоит из:

- ввода значений ***x*** и ***y***;
- вычисления значения ***t1***;
- вычисления значения ***t2***;
- вывода значений ***t1*** и ***t2***.

Оптимизация алгоритма

Перед непосредственным программированием алгоритма проанализируем, как в нем можно изменить объем вычислений.

Выражение ***ax*** встречается один раз в первой формуле и дважды - во второй. Следовательно, можно один раз произвести умножение ***a*x***, а потом использовать этот результат.

Во второй формуле дважды встречается тангенс - это вычисление можно так же сделать один раз.

Ограничения на значения параметров

Пример ограничений:

Аргумент функции, которую вычисляет логарифм, не может быть 0 или меньше.
Аргумент функции извлечения квадратного корня не может быть меньше 0
В знаменателе выражения не может быть 0, отсюда:

Определение переменных программы

Для решения задачи нам понадобятся переменные для представления каждого параметра формул - ***a, b, x, y*** и результатов - ***t1, t2***. Кроме того, придется ввести дополнительную переменную ***ax*** для хранения промежуточного результата , необходимого для оптимизации. Тип всех переменных - ***double.***

Разработка текста программы

Программа начинается с включения файлов:

```
#include <stdio.h>
#include <math.h>
```

в которых находятся описания функций ввода - вывода и математических функций соответственно.

Далее открываем главную функцию:

```
int main(void)
```

Включаем описания переменных:

```
double x,y;
   double a=12.5,  b=1.3;
   double c=14.1,  d=2.7;
   double t1, t2;
   double ax;
```

Вводятся значения для переменных ***x*** и ***y***:

```
   printf("Введите x, y >");
   scanf("%lf %lf",&x,&y);
```

Далее вычисляются промежуточные и окончательные результаты, после чего полученные результаты выводятся на экран:

```
   printf("t1 = %lg\n",t1); printf("t2 = %lg\n",t2);
```

Отладка программы

Задание 2. При отладке программы можно проверять правильность выполнения каждой операции. Для этого сложные операторы-выражения, разбиваются на последовательность операторов-выражений, в каждом из которых выполняется только одна операция. Результат каждой такой операции выводится на экран или отслеживается в пошаговом режиме.

Результаты работы программы

При работе программы на экран было выдано следующее:

```
Введите x, y >3.3 1.1
t1 = 0.348897
t2 = 0.0133405
```