Балтийский государственный технический университет  
«ВОЕНМЕХ» им. Д. Ф. Устинова

Кафедра О7 «Информационные системы и программная инженерия»

**Практическая работа №1**по дисциплине «Информатика: Основы программирования»  
на тему «Структура программы, основные типы данных, ввод/вывод»

Выполнил:  
Студент *Фамилия И.О.*  
Группа *Номер\_группы*  
  
Преподаватель:  
*Фамилия И.О.*

Санкт-Петербург  
2022 г.

**Задание 1.**

Написать программу, которая будет находить сумму любых двух целых чисел, введенных с клавиатуры.

*Входные данные:* слагаемые, два целых числа. Обозначим их a и b, тип int.

*Выходные данные:* сумма, целое число. Обозначим как s, тип int.

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
| а = 2, b = 2 | 4 |  |
| а = 2000, b = -2000 | 0 |  |
| а = 2000000000, b = 2000000000 | 4000000000 |  |

Текст программы:

#include <stdio.h>

#include <stdlib.h>

int main()

{

int a, b, s; /\* объявление переменных \*/

printf ("a = "); /\* печать сообщения \*/

scanf ("%d", &a); /\* ввод с клавиатуры вещественного числа и запись его в переменную a \*/

printf ("b = "); /\* печать следующего сообщения \*/

scanf ("%d", &b); /\* ввод с клавиатуры вещественного числа и запись его в переменную b \*/

s = a + b; /\* вычисление суммы и запись ее в переменную s \*/

printf ("%d + %d = %d\n", a, b, s); /\* вывод результата в формате число + число = число \*/

return 0;

}

*Выводы*: *здесь пишем свои выводы*.

**Задание 2.**

Написать программу деления одного целого числа на другое.

*Входные данные*: *описываем выходные данные, их обозначение в программе,* тип int.

*Выходные данные*: *описываем выходные данные, их обозначение в программе* тип int.

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
| *a = 4, b = 2* | *2* |  |
| *a = 7, b = 3* | *2,33* |  |
| *a = 5, b = 3* | *1,67* |  |
| *a = 1, b = 2* | *0,5* |  |
| *a = 5, b = 0* | *Сообщение об ошибке* |  |

Текст программы:

*Сюда добавляем текст программы. Шрифт Courier New или FreeMono 10 пт, междустрочный интервал одинарный.*

Выводы: *здесь пишем свои выводы*.

**Задание 3.**

Изменить тип переменных в предыдущей программе на *double* (стандартный вещественный тип). В функциях *scanf()* и *printf()* поменять спецификаторы формата на *%lf*.

*Входные* и *выходные данные* те же, что и в задании 2, обозначения переменных те же, тип всех переменных double.

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
| *a = 4, b = 2* | *2* |  |
| *a = 7, b = 3* | *2,33* |  |
| *a = 5, b = 3* | *1,67* |  |
| *a = 1, b = 2* | *0,5* |  |
| *a = 5, b = 0* | *Сообщение об ошибке* |  |
| *a = 4.2, b = 2.1* | *2* |  |
| *a = 5.5, b = 2.2* | *2,5* |  |
| *a = 4.4, b = 0.1* | *44* |  |

Текст программы:

*Сюда добавляем текст программы. Шрифт Courier New или FreeMono 10 пт, междустрочный интервал одинарный.*

Выводы: *здесь пишем свои выводы*.

При изменении формата вывода на *%.8lf* выводимое значение стало таким: *значение*.

При изменении формата вывода на *%.2lf* выводимое значение стало таким: *значение*.

Выводы: *здесь пишем свои выводы*.

**Задание 4.**

В предыдущей программе изменить тип делимого и делителя обратно на int, результат оставить типа double.

*Входные данные*: *описываем выходные данные, их обозначение в программе,* тип int.

*Выходные данные*: *описываем выходные данные, их обозначение в программе,* тип double.

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
|  |  |  |
|  |  |  |
|  |  |  |

Текст программы:

*Сюда добавляем текст программы. Шрифт Courier New или FreeMono 10 пт, междустрочный интервал одинарный.*

Выводы: *здесь пишем свои выводы*.

**Задание 5.**

Проанализировать ошибки при вызове функций scanf() и printf().

|  |  |
| --- | --- |
| **Ошибка** | **Поведение программы** |
| отсутствие & перед именем переменной в scanf() |  |
| наличие & перед именем переменной в printf() при выводе значения переменной |  |
| тип спецификатора формата ввода не совпадает с типом переменной: переменная типа *int*, спецификатор *%lf* |  |
| тип спецификатора формата ввода не совпадает с типом переменной: переменная типа *double*, спецификатор *%d* |  |
| тип спецификатора формата ввода не совпадает с типом переменной: переменная типа *double*, спецификатор *%f* |  |
| тип спецификатора формата вывода не совпадает с типом значения: значение типа *int*, спецификатор *%lf* |  |
| тип спецификатора формата вывода не совпадает с типом значения: значение типа *double*, спецификатор *%d* |  |
| количество спецификаторов формата ввода меньше количества вводимых значений переменных |  |
| количество спецификаторов формата ввода больше количества вводимых значений переменных |  |
| количество спецификаторов формата вывода меньше количества выводимых значений |  |
| количество спецификаторов формата вывода больше количества выводимых значений |  |

**Задание 6.**

Познакомьтесь с типами данных *char* и *unsigned char*.

#include <stdlib.h>

#include <stdio.h>

#include <limits.h> */\* комментарии \*/*

int main()

{

char c;

unsigned char uc;

printf("sizeof(c)=%d\tsizeof(uc)=%d\n\n", sizeof(c),sizeof(uc)); */\* комментарии \*/*

uc = c = CHAR\_MAX; */\* комментарии \*/*

printf("CHAR\_MAX : c=%d uc=%d\n", c, uc); */\* комментарии \*/*

c = c + 1; uc = uc + 1; */\* комментарии \*/*

printf("CHAR\_MAX+1 : c=%d uc=%d\n", c, uc);

uc = c = CHAR\_MIN; */\* комментарии \*/*

printf("CHAR\_MIN : c=%d uc=%d\n", c, uc);

c = uc = UCHAR\_MAX; */\* комментарии \*/*

printf("UCHAR\_MAX : c=%d uc=%d\n", c, uc);

c = c + 1; uc = uc + 1; */\* комментарии \*/*

printf("UCHAR\_MAX+1 : c=%d uc=%d\n", c, uc);

uc = c = -5; */\* комментарии \*/*

printf("-5 : c=%d uc=%d\n", c, uc);

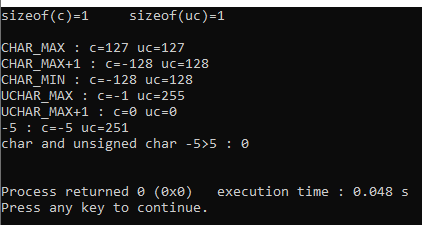
c = -5; uc = 5; */\* комментарии \*/*

printf("char and unsigned char -5>5 : %d\n\n", c>uc); */\* комментарии \*/*

return 0;

}

Результаты работы программы:



**Задание 7.**

Познакомьтесь с типами данных *int, short int, long int* и *unsigned int*.

#include <stdlib.h>

#include <stdio.h>

#include <limits.h>

int main()

{

char c;

unsigned char uc;

int i;

unsigned u;

short s;

long l;

printf("sizeof(i)=%d\tsizeof(u)=%d\tsizeof(s)=%d\tsizeof(l)=%d\n\n",

sizeof(i), sizeof(u), sizeof(s), sizeof(l)); */\* комментарии \*/*

c = s = SHRT\_MAX; */\* комментарии \*/*

uc = s; */\* комментарии \*/*

printf("SHRT\_MAX : c=%d uc=%d s=%d\n", c, uc, s);

s = s + 1; */\* комментарии \*/*

printf("SHRT\_MAX+1 : s=%d\n", s);

c = s; uc = s; */\* комментарии \*/*

printf("%d : c=%d uc=%d\n", SHRT\_MIN, c, uc);

s = 0; c = s; uc = s; */\* комментарии \*/*

printf("0 : c=%d uc=%d s=%d\n", c, uc, s);

i = INT\_MAX; */\* комментарии \*/*

l = i; u = i; */\* комментарии \*/*

printf("INT\_MAX : i=%d u=%u l=%ld\n", i, u, l); */\* комментарии \*/*

i = i + 1; l = l + 1; u = u + 1; */\* комментарии \*/*

printf("INT\_MAX+1 : i=%d u=%u l=%ld\n", i, u, l);

i = INT\_MIN; */\* комментарии \*/*

l = i; u = i; */\* комментарии \*/*

printf("INT\_MIN : i=%d u=%u l=%ld\n", i, u, l);

u = UINT\_MAX; */\* комментарии \*/*

i = u; l = u; */\* комментарии \*/*

printf("UINT\_MAX : i=%d u=%u l=%ld\n", i, u, l);

u = i = -5; */\* комментарии \*/*

printf("-5 : i=%d u=%u\n", i, u);

i = -5; u = 5; */\* комментарии \*/*

printf("int and unsigned int -5>5 : %d\n", i > u);

c = -5; u = 5; */\* комментарии \*/*

printf("char and unsigned int -5>5 : %d\n\n", c > u);

i = 5.1; */\* комментарии \*/*

printf("i=5.1 : i=%d\n", i);

i = 5.9; */\* комментарии \*/*

printf("i=5.9 : i=%d\n", i);

return 0;

}

Результаты работы программы:

*скриншот*

**Задание 8.**

Познакомьтесь с типами данных *float* и *double*.

#include <stdlib.h>

#include <stdio.h>

#include <float.h> */\* комментарии \*/*

int main()

{

float f;

double d;

printf("sizeof(f)=%d\tsizeof(d)=%d\n\n", sizeof(f), sizeof(d)); */\* комментарии \*/*

d = f = FLT\_MAX; */\* комментарии \*/*

printf("FLT\_MAX : f=%g d=%g\n", f, d);

d = f = FLT\_MIN; */\* комментарии \*/*

printf("FLT\_MIN : f=%g d=%g\n", f, d);

d = f = FLT\_EPSILON; */\* комментарии \*/*

printf("FLT\_EPSILON : f=%g d=%g\n", f, d);

f = 12345678; */\* комментарии \*/*

printf("12345678 : f=%f\n", f);

f = 123456789; */\* комментарии \*/*

printf("123456789 : f=%f\n", f);

f = 1234567890; */\* комментарии \*/*

printf("1234567890 : f=%f\n", f);

d = DBL\_MAX; */\* комментарии \*/*

printf("DBL\_MAX : d=%g\n", d);

d = DBL\_MIN; */\* комментарии \*/*

printf("DBL\_MIN : d=%g\n", d);

d = DBL\_EPSILON; */\* комментарии \*/*

printf("DBL\_EPSILON : d=%g\n", d);

d = 1e15 + 1; */\* комментарии \*/*

printf("1e15+1 : d=%lf\n", d);

d = 1e16 + 1; */\* комментарии \*/*

printf("1e16+1 : d=%lf\n", d);

d = 10000 \* 100000 + 1 - 4 \* 250000000; */\* комментарии \*/*

printf("1 : d=%lf\n", d);

d = 1e20 \* 1e20 + 1000 - 1e22 \* 1e18; */\* комментарии \*/*

printf("1000 : d=%lf\n", d);

d = 1e20 \* 1e20 - 1e22 \* 1e18 + 1000; */\* комментарии \*/*

printf("1000 : d=%lf\n", d);

f = d = 0.3; */\* комментарии \*/*

printf("0.3 : f=%.8f d=%.17f\n", f, d);

f = 0;

while (f < 10) */\* комментарии \*/*

f += 0.2;

printf("10 : f=%f\n", f);

return 0;

}

Результаты работы программы:

*скриншот*

**Задание 9.**

Проверить порядок выполнения операций в каждом выражении, содержащем несколько операций присваивания, разделив каждый оператор-выражение на несколько операторов, выполняемых последовательно.

Текст измененной программы:

*Сюда добавляем текст программы. Шрифт Courier New или FreeMono 10 пт, междустрочный интервал одинарный.*

Результаты работы программ:

до изменения

*скриншот*

после изменения

*скриншот*

**Задание 10.**

Написать программу для вычисления значений следующих выражений:  
a=5, c=5

a=a+b-2

c=c+1, d=c-a+d

a=a\*c, c=c-1

a=a/10, c=c/2, b=b-1, d=d\*(c+b+a)

Выражения, записанные в одной строке, записывать одним оператором-выражением, не содержащим запятой. Использовать расширенные операции присваивания, операции инкремента и декремента. Переменные c и d объявить как целые, переменные a и b – как вещественные. Значения переменных b и d вводить с клавиатуры. После вычисления каждого выражения выводить на экран значения всех переменных.

Таблица тестирования:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Ожидаемый результат** | **Результат работы программы** |
|  |  |  |
|  |  |  |
|  |  |  |

Текст программы:

*Сюда добавляем текст программы. Шрифт Courier New или FreeMono 10 пт, междустрочный интервал одинарный.*