Лабораторная работа 2

**Знакомство с языком С++: команды ввода-вывода, математические выражения**

Команды ввода/вывода, наследуемые С++ из языка Си

Вывод данных **printf\_s()** Ввод данных **scanf\_s()**

Функция printf\_s() и scanf\_s()имеют прототип в файле **stdio.h, поэтому его необходимо подключить** до основной функции main()

Пример:

#include "stdio.h"

int main()

{

int x;

printf\_s("Input X:");

scanf\_s("%d", &x);

int y = x;

printf\_s("y=%d", y);

return 0;

}

**Функция printf\_s()**

Функция форматированного вывода, то есть в параметрах функции необходимо указать формат данных, которые будут выводиться. Формат данных указывается спецификаторами формата, который начинается с символа % и заканчивается кодом формата.

**Спецификаторы формата:**

|  |  |
| --- | --- |
| %с | символ |
| %d | целое десятичное число |
| %i | целое десятичное число |
| %e | десятичное число в виде x.xx e+xx |
| %E | десятичное число в виде x.xx E+xx |
| %f | десятичное число с плавающей запятой xx.xxxx |
| %F | десятичное число с плавающей запятой xx.xxxx |
| %g | %f или %e, что короче |
| %G | %F или %E, что короче |
| %o | восьмеричное число |
| %s | строка символов |
| %u | беззнаковое десятичное число |
| %x | шестнадцатеричное число |
| %X | шестнадцатеричное число |
| %% | символ % |
| %p | Указатель |
| %n | Указатель |

Кроме того, к командам формата могут быть применены модификаторы l и h.

|  |  |
| --- | --- |
| %ld | печать long int |
| %hu | печать short unsigned |
| %Lf | печать long double |

После символа % может быть указана точность (число цифр после запятой). Задание точности: %.n<код формата>, где n – число цифр после запятой, а <код формата> – один из кодов приведённых выше.

**Примеры:**

1. Вывести x **с точностью** до 3-х цифр после запятой

float x = 10.3563;

printf\_s("Переменная x = %.3f", x);

Результат: x = 10.356

1. Указание минимальной **ширины** поля для печати:

printf\_s("%5d", 20);

Результат: 20

1. Вместо ширины и точности в спецификаторе формата можно указывать символ \* (звездочка). Это позволяет задавать и ширину, и точность как значения целых переменны.

// Переменные ширина и точность

float number = -12.3;

int w = 10, p = 4; // ширина и точность

printf\_s("%0\*.\*f\n", w, p, number);

w = 15, p = 5; // новая ширина и точность

printf\_s("%0\*.\*f\n", w, p, number);

Результат:

-0012.3000

-00000012.30000

**Функция scanf\_s()**

Форматированный ввод. С её помощью можно вводить данные со стандартного устройства ввода (клавиатуры). Вводимыми данными могут быть целые числа, числа с плавающей запятой, символы, строки и указатели.

Прототип функции: int scanf\_s(char \**управляющая строка*);

Функция возвращает число переменных, которым было присвоено значение.

*Управляющая строка* содержит три вида символов:

спецификаторы формата начинаются с символа %.,

пробелы

другие символы.

**Спецификаторы формата:**

|  |  |
| --- | --- |
| %c | чтение символа |
| %d | чтение десятичного целого |
| %i | чтение десятичного целого |
| %e | чтение числа типа float (плавающая запятая) |
| %h | чтение short int |
| %o | чтение восьмеричного числа |
| %s | чтение строки |
| %x | чтение шестнадцатеричного числа |
| %p | чтение указателя |
| %n | чтение указателя в увеличенном формате |

**Пример:**

Вывести x **с точностью** до 3-х цифр после запятой

float x=0, a=0;

int result = scanf\_s("%f %f", &x, &a);

printf\_s("input %f\n", a);

printf\_s("input %f\n", x);

double c = 0.0;

c=pow(a, 3);

printf\_s("c=%.3f", c);

Результат: с = 8

**Замечание:** при вводе строки с помощью функции scanf\_s() (спецификатор формата %s), строка вводится до первого пробела.

Пример: ввести строку "Привет мир!" с использованием функции scanf\_s()

char str[80]; // массив на 80 символов

int result = scanf\_s("%s", str, 79);

Результат: в массиве str будет храниться одно слово "Привет".

Для введения всей строки используют функцию gets\_s (), которая читает символы с клавиатуры до появления символа новой строки (\n), то есть до нажатия клавиши enter.

Функция возвращает указатель на buf, где buf - буфер (память) для вводимой строки.

**Пример:**

char buffer[100]; // массив (буфер) для вводимой строки

gets\_s(buffer, 99); // ввод строки до нажатия enter

printf\_s("%s", buffer); // вывод введённой строки на экран

**Замечание:** для ввода данных с помощью функции scanf\_s(),в качестве параметров ей нужно передавать **адреса переменных**, а не сами переменные.

Получить адрес переменной – установить перед именем переменной знак &(амперсанд). Знак & означает взятие адреса (адрес в памяти компьютера где хранится значение переменной.)

**Математические выражения**

В заголовочном файле <math.h> определены функции, выполняющие некоторые часто используемые математические задачи:

| **Функция** | **Описание** | **Пример** |
| --- | --- | --- |
| **abs( a )** | модуль или абсолютное значение от **а** | abs(-3.0)= 3.0 abs(5.0)= 5.0 |
| **sqrt(a)** | корень квадратный из **а,**причём **а**не отрицательно | sqrt(9.0)=3.0 |
| **pow(a, b)** | возведение **а**в степень **b** | pow(2,3)=8 |
| **ceil( a )** | округление **а** до наименьшего целого, но не меньше чем **а** | ceil(2.3)=3.0 ceil(-2.3)=-2.0 |
| **floor(a)** | округление **а** до наибольшего целого, но не больше чем **а** | floor(12.4)=12 floor(-2.9)=-3 |
| **fmod(a, b)** | вычисление остатка от  a/b | fmod(4.4, 7.5) = 4.4 fmod( 7.5, 4.4) = 3.1 |
| **exp(a)** | вычисление экспоненты **еа** | exp(0)=1 |
| **sin(a)** | **a** задаётся в радианах |  |
| **cos(a)** | **a** задаётся в радианах |  |
| **log(a)** | натуральный логарифм **a**(основанием является экспонента) | log(1.0)=0.0 |
| **log10(a)** | десятичный логарифм **а** | Log10(10)=1 |
| **asin(a)** | арксинус **a**, где **-1.0 < а < 1.0** | asin(1)=1.5708 |

|  |
| --- |
| **Полезная информация**   1. Вывод текста на русском языке   Подключаем к проекту заголовочный файл <locale.h> или <clocale>  #include <clocale>  В начале программы вызываем функцию  setlocale(LC\_CTYPE, "rus");  или  setlocale(LC\_ALL, "Russian");  или  setlocale(0, "");   1. Задержка экрана   Подключаем к проекту заголовочный файл <conio.h>  В конце программы вызываем функцию \_getch();  #include <conio.h>  {  ...  \_getch();  return 0;  } |

***Задачи для самостоятельного решения:***

**Типы данных: int, float, double**

1. Используя оператор sizeof, выведите на экран размер следующих типов данных: bool, char, wchar\_t, int,short int, long int, long long int, float, double, long double.
2. Найдите значение функции *y*= |*x*5+ |*ax* – *x*3| – *a*| + *ax*2+ *a*8 для любых целых значений *a* и *x*.
3. Дано трехзначное число. Получите четырехзначное число, приписав цифру единиц в качестве цифры тысяч (345 -> 5345).
4. С начала суток прошло N секунд. Вывести заданное время в виде чч:мм:сс.
5. Точки A и B заданы координатами на плоскости: A(x1, y1), B(x2, y2). Найти:
   1. длину отрезка AB;
   2. координаты середины отрезка AB.
6. Запишите выражение на языке программирования:
   1. 
   2. 

Вычислите значения для заданных с клавиатуры *x*. В ответе отведите 4 знака под дробную часть.

1. С клавиатуры вводятся значения углов треугольника в градусах и радиус описанной около него окружности. Найдите стороны треугольника. Используйте математическую константу M\_PI для перевода градусов радианы.

#define \_USE\_MATH\_DEFINES

#include <math.h> // число pi