Стандартные функции ввода-вывода

- Функции **printf** (для вывода) и **scanf** (для ввода) позволяют преобразовывать численные величины в символьное представление и обратно. Функции описаны следующим образом:
- **int printf(const char *format, ...); int scanf(const char *format, ...);**

Стандартные функции ввода-вывода

- Функция **printf** возвращает количество выведенных символов (значение **EOF** свидетельствует об ошибке). Функция **scanf** возвращает количество значений, успешно присвоенных указанным переменным.
- Первый аргумент функций printf и scanf представляет собой **управляющую (форматную) строку**, которая указывает, в каком виде следует выводить или вводить последующие аргументы.

Стандартные функции ввода-вывода

- Управляющая строка содержит два типа объектов: обычные символы и **спецификации преобразований**.
- Каждая спецификация преобразования начинается с **символа %** и заканчивается символом преобразования.
- Символы преобразования: **d** – десятичная форма, **o** – восьмеричная форма, **x** – шестнадцатеричная форма, **u** – беззнаковая десятичная форма, **c** – отдельный символ, **s** – строка, **e** – число в научном формате (с экспонентой), **f** – число с плавающей запятой.

Стандартные функции ввода-вывода

- В спецификациях преобразований функции **printf** между **знаком %** и **символом преобразования** могут находиться:
- **число, задающее ширину поля,**
- **точка-разделитель,**
- **число, задающее точность представления,**
- **модификатор длины l для данных типа long**

Стандартные функции ввода-вывода

- Функции **printf и scanf** могут иметь произвольное количество аргументов.
- **Аргументы**, следующие за управляющей строкой, для функции **printf** представляют собой подлежащие **выводу значения переменных и выражений**, а для функции **scanf - указатели на переменные**, в которых должны быть сохранены введенные значения.

Стандартные функции ввода-вывода

- Таким образом, перед **каждым** аргументами функции **scanf**, который является **именем переменной**, должен стоять амперсанд – символ «**&**». Если вводится строка, то **перед именем** массива амперсанд **не нужен**, так как **имя массива** само по себе является **указателем**.

Стандартные функции ввода-вывода

- Приведем в качестве примера программу, которая вводит значение N и печатает таблицу квадратов чисел от 1 до N:
- **#include <stdio.h> void main(void**
- **int i, N;**
- **printf(”Введите N: ”);**
- **scanf(”%d”, &N);**
- **printf(”Таблица квадратов:\n”);**
- **for(i=1; i<=N; i++) printf(”%2d %4d\n”, i, i*i);**
- **}**

Стандартные функции ввода-вывода

- Для того, чтобы колонки **таблицы выводились ровно,** требуется **задать** для каждого выводимого **значения ширину поля**. В данном случае ширина первой колонки – **два** знакоместа, ширина второй – **четыре**.
- В качестве аргументов функции **printf** можно использовать выражения. В приведенном примере с помощью **выражения** вычисляется квадрат **числа i.**

Стандартные функции ввода-вывода

- Рассмотрим пример другой программы, которая использует цикл **while** для вычисления **длины строки**:
- **#include <stdio.h>**
- **void main(void)**
- **{**
- **int i;**
- **char s[100];**
- **printf(”Введите строку:\n”);**
- **gets(s);**
- **i=0;**
- **while(s[i] != 0) i++;**
- **printf(”Длина строки %d символов.\n”);**
- **}**

Стандартные функции ввода-вывода

- В следующем примере по радиусу круга вычисляется его площадь:
- **#include <stdio.h> main()**
- **{**
- **double r, pi=3.1415926535;**
- **printf("Введите радиус круга: "); scanf("%lf", &r);**
- **printf("Площадь круга: %f\n", pi*r*r);**
- **}**

Стандартные функции ввода-вывода

- В стандарте языка **Си** отсутствует специальное обозначение для числа **p** и его приходится задавать в виде **вещественной константы**.
- При выводе чисел типа **double** использовать в спецификации преобразования признак удвоенной точности (букву **l**) не обязательно. Однако при вводе значений типов **long** и **double** такой признак в спецификация **необходим**, иначе значение будет введено неверно

Математические функции

- Математические функции Си описаны в заголовочном файле math.h. Программа, которая использует математические функции, должна содержать следующую строку:
- **#include <math.h>**

Математические функции

- Наиболее часто применяются следующие функции:
- **double cos(double x)** – вычисляет косинус аргумента;
- **double fabs(double x)** – вычисляет абсолютное значение аргумента;
- **double exp(double x)** – вычисляет экспоненциальную функцию аргумента;
- **double log(double x)** – вычисляет натуральный логарифм аргумента;
- **double log10(double x)** – вычисляет десятичный логарифм аргумента;
- **double sin(double x)** – вычисляет синус аргумента;
- **double sqrt(double x)** – вычисляет квадратный корень аргумента;
- **double tan(double x)** – вычисляет тангенс аргумента.

Математические функции

- Из приведенных описаний видно, что **математические функции** работают с **вещественными числами** удвоенной **точности:** аргументы имеют типа **double** и возвращаемое значение также имеет тип **double**.
- В качестве примера рассмотрим программу, которая вводит значение ***x***, вычисляет значение ***exp(x)*** и выводит полученное значение на экран:

Математические функции

- #include <stdio.h> #include <math.h> main()
- {
- double x, y;
- printf("Введите значение x: "); scanf("%lf", &x);
- y = exp(x); printf("exp(x)=%f\n", y);
- }