Санкт-Петербургский колледж информационных технологий

Основы программирования.
Типы данных

Учебное пособие по курсу
«Основы программирования»

Преподаватель Алексеева Н.Н.

Санкт-Петербург
2011

Типы данных

Различают типы данных и модификаторы типов.

Базовые типы: char, int, float, double, void

Тип	Описание	Размер	Диапазон значений
char	символьный	1	256 значений символов
int	целый	2	-32768… 32767
float	вещественный	4	1,17E-38…3,37E+38
double	веществ. с дв. точностью	8	2,23E-308…1,67E+308

Модификаторы целого типа:

Модифик.	Описание	Размер	Диапазон значений
unsigned int	беззнаковый	2	0…65535
signed int	знаковый	2	-32768… 32767
signed short int	короткий	2	-32768… 32767
unsigned short int		2	0…65535
signed long int	длинный	4	-2147483648…2147483647
unsigned long int		4	0…4 294 967 295

Операция typedef

Операция typedef позволяет переопределить тип переменной, дав ему новой имя.

typedef тип новое_имя;

Пример:

```
#include <iostream.h>
typedef  unsigned short int  USINT;
void main()
{
 USINT i=24, j=18;
cout << i*j;
getch();
}
```

Операция приведения типа

Операция приведения типов позволяет переопределить тип переменной во время выполнения арифметических операций.

Пример:

```
#include <iostream.h>
void main()
{
 int i=10, j=3, a;
a=i/j;
cout << a;

float a=(float) i/ (float) j;
cout << a;
getch();
}
```

Функции ввода- вывода

Для использования функции ввода-вывода данных PRINTF и SCANF необходимо подключить библиотеку stdio.h директивой include.

Функция printf ()

printf (*fmt_string, ...);*

Первый аргумент, *fmt_string,* определяет способ отображения всех последующих аргументов. Этот аргумент часто называют *строкой форматирования:*

% [флаг] [ширина] [точность] [модификатор_типа] тип_формата

Типы формата перечислены в таблице. Количество аргументов должно в точности совпадать с количеством команд форматирования, причем совпадение обязательно и в порядке их следования.

Пример:

printf ("Привет %c %d %s", 'c', 10, "всем!"); Результат: Привет c 10 всем!

printf (“\n k=%d t=%d”, k, t); Результат: k=0 t=0

printf (“\nПовторить? (y/n)"); Результат: Повторить? (y/n)

Функция вывода PRINTF

Флаг – определяет выравнивание выводимых данных.

Знак	Действие
-	Выравнивание по левому краю
+	Выводит знак числа
пробел	Выводит знак пробела перед числом
0	Заполняет поле нулями

Например, строка форматирования `%05d` дополнит выводимое число нулями (их будет меньше пяти), чтобы общая длина была равной пяти символам.

Ширина - задает минимальную ширину поля в символах. Если выводимое значение (строка или число) больше этого минимума, оно будет выведено полностью, несмотря на превышение минимума.

Точность - количество десятичных знаков, выводимых после точки.

Модификаторы:

h – short

I – long

L – long double

Функция вывода PRINTF

Тип формата	Формат
%c	Символ
%d (%i)	Десятичное целое со знаком
%u	Десятичное целое без знака
%e(%E)	Экспоненциальное представление -строчная буква e (E): [-]d.dddde[+/-]ddd ([-]d.ddddE[+/-]ddd)
%f	Значение с плавающей точкой
%g	Использует более короткий из двух форматов: **%f** или **%g**
%o	Восьмеричное целое без знака
%s	Строка символов
%x	Шестнадцатеричное целое без знака
%p	Указатель
%%	Выводит символ %

Функция вывода PRINTF

Чтобы добавить модификатор точности, поставьте за спецификатором ширины поля десятичную точку, а после нее — значение спецификации точности. Для форматов d, D, e, E, f и F модификатор точности определяет число выводимых десятичных знаков.

Например, строка форматирования %10 . 4f обеспечит вывод числа, ширина которого составит не меньше десяти символов, с четырьмя десятичными знаками.

Применительно к целым или строкам, число, следующее за точкой, задает максимальную длину поля. Например, строка форматирования %5 . 7 s отобразит строку длиной не менее пяти, но не более семи символов. Если выводимая строка окажется длиннее максимальной длины поля, конечные символы будут отсечены.

Например, строка форматирования %-10.2f обеспечит выравнивание вещественного числа (с двумя десятичными знаками в 10-символьном поле) по левому краю.

Пример:

```
Long double x=123.4567;
printf (“ \n x1=%Lf\n x2=%+.2f \n x3=%.0f”, x, x, x);
```

Функция вывода PRINTF

Пример:

```
#include <stdio.h>
#include <conio.h>
void main()
{
printf ("|%11.6f|\n", 12 3.23);
printf ("|%-11.6f|\n", 123.23);
printf ("|%11.6s|\n", "Привет всем");
printf("I%-11.6s I\n", "Привет всем");
}
```

При выполнении эта программа отображает такие результаты.

```
|  123.230000|
|123.230000  |
|    Привет 1 |
|Привет       |
```

Операция sizeof

Операция sizeof определяет размер памяти, который соответствует идентификатору или типу.

sizeof (имя)

Пример:

```
#include <stdio.h>
#include <conio.h>
void main()
{
 int i, j, k, l, m; float x; char c;
i=sizeof(i);  j=sizeof(c);  k=sizeof(x);  l=sizeof(double);  m=sizeof(long double);
 printf ("\n size int=%d, char=%d, float=%d, doble=%d, long double=%d\n", i, j, k, l, m);
getch();
}
```

Функции ввода scanf ()

Фннкция **scanf ()** используется для ввода данных в программу с клавиатуры.

scanf (*fmt_string, ...);*

Управляющая строка, задаваемая параметром *fmt_string,* состоит из символов трех категорий:

- спецификаторов формата;
- "пробельных" символов (пробелы, символы табуляции и пустой строки);
- символов, отличных от "пробельных".

Все переменные, используемые для приема значений с помощью функции scanf (), должны передаваться посредством их адресов. Это значит, что все аргументы должны быть указателями на переменные (перед именем переменной необходимо поставить значок &).

Пример:

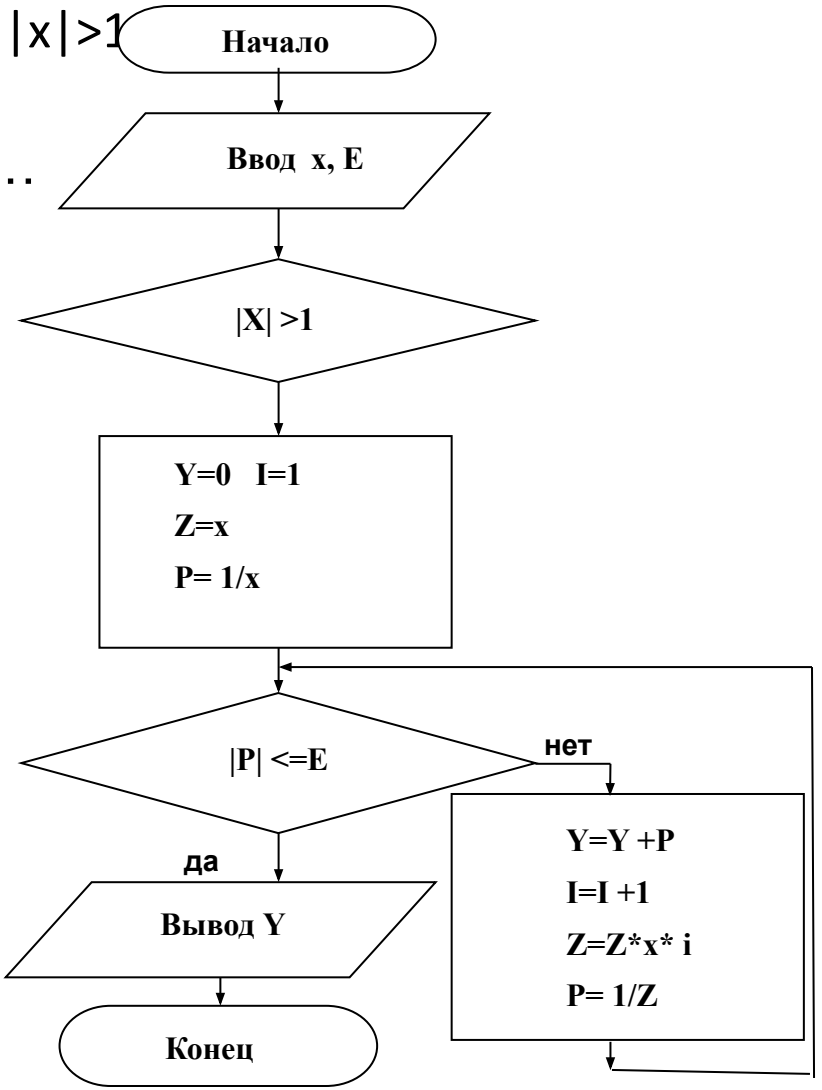
```
scanf("%d%  %d",  &x,  &y);
scanf("%20s",  str);
```

Вычисление суммы ряда

Задание: Вычислить сумму $\sum P_i$ с заданной точностью E. Считать, что заданная точность достигнута, если отдельное слагаемое по модулю стало меньше E, т.е. $|P_i| < E$. ($|x| > 1$

$$Y = 1/x*1! + 1/x^2*2! + 1/x^3*3! + 1/x^4*4! + \ldots$$

Блок-схема алгоритма:



Вычисление суммы ряда

Программа:

```
#include <stdio.h>
#include <iostream.h>
#include <conio.h>

void main()
{
int  i;
float x, y, z, p, e;
clrscr();
e=0.0001;
cin>>x;
p=x; z=x; i=1; y=1/x;
while(p>e)
 {  i++;
    z=z*x*i;
    p=1/z;
    y=y+p;
 }
printf("cymma c toch %e ravna %f",e,y);
getch();
}
```

Вычисление суммы ряда – 2 способ

Вычисление продолжается до тех пор, пока разность между текущим и следующим элементом ряда не станет меньше заданной точности E.

```
void main()
{
int n,k,i,d;  float x,y,z,p0,p1,e;
clrscr();
e=0.0001;
cin>>x;
i=1;
z=x; p0=1/z; y=p0;
i++;
z=z*x*i; p1=1/z; y=y+p1;
while((p0-p1)>e)
 {  p0=p1;
    i++;
    z=z*x*i;    p1=1/z;    y=y+p1;
 }
printf("cymma c toch %e ravna %f",e,y);
getch();
}
```

Задание

Вариант	Сумма членов ряда	Значение x	Точность вычисления
1	$s = -\frac{(2x)^2}{2} + \frac{(2x)^4}{24} + \ldots + (-1)^n \frac{(2x)^{2n}}{(2n)!} + \ldots$	0,20	10^{-5}
2	$s = x - \frac{x^3}{3} + \ldots + (-1)^n \frac{x^{2n-1}}{2n-1} + \ldots$	0,10	$0{,}5 \cdot 10^{-4}$
3	$s = \frac{x^3}{5} - \frac{x^5}{17} + \ldots + (-1)^{n+1} \frac{x^{2n+1}}{4n^2+1} + \ldots$	0,15	10^{-3}
4	$s = \frac{x^2}{2!} + \frac{x^4}{4!} + \ldots + \frac{x^{2n}}{(2n)!} + \ldots$	0,70	10^{-4}
5	$s = \frac{1}{x} - \frac{1}{3x^3} + \frac{1}{5x^5} - \ldots + (-1)^n \frac{1}{(2n+1)x^{2n+1}}$	1,5	$0{,}5 \cdot 10^{-3}$
6	$s = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \ldots + \frac{x^{2n+1}}{(2n+1)!} + \ldots$	1,7	10^{-3}
7	$s = 1 + \frac{x^2}{2!} - \frac{3x^4}{4!} + \ldots + (-1)^n \frac{2n-1}{(2n)!} x^{2n} + \ldots$	0,75	$0{,}5 \cdot 10^{-3}$
8	$s = \frac{x^3}{3} + \frac{x^5}{15} + \ldots + (-1)^{n+1} \frac{x^{2n+1}}{4n^2-1} + \ldots$	0,30	10^{-5}