

ПРОГРАММИРОВАНИЕ. РАБОТА С СИМВОЛАМИ

Действия со строками

- Конкатенация – сложение (“склеивание”) строк
 - Пример: $a='ab'$, $b='xy'$, $c=a+b='abxy'$
- Сравнение строк:
 - Пример: $'abc' < 'abd'$, $'Abc' < 'abc'$, $'bc' > 'ab'$

Теоретический материал для pascal

- **PROGRAM** < ИМЯ >;
- **CONST ...** описание констант;
- **VAR ...** описание переменных; //глобальные переменные
- **FUNCTION** <имя>(аргумент функции:тип аргумента):
тип функции
- **begin**
- }
- **END.**

Операции

- X:=5; // x присвоить 5
- Readln(x); // ввести x
- Writeln (x); // вывести x
- If (y) then (д1) else (д2); // условие: Если условие (y) выполняется, тогда делаем действие (д1), иначе действие (д2)
- While (y) do // пока условие в скобках истинно, выполняется тело подпрограммы
- for <пар.цикла> := <нач.знач> to <кон.знач.> do тело цикла// для параметра цикла целочисленного типа с начальным значением до конечного значения делаем (д)

Основные функции

- **Copy(S, poz, n)** - выделяет из строки S, начиная с позиции poz, подстроку из n символов;
- **Length(S)** определяет текущую длину строкового выражения S;
- **Pos(subS, S)** определяет позицию первого вхождения подстроки subS в строку S;
- **Delete (S, poz, n)** удаляет из строки S, начиная с позиции poz, подстроку из n символов;
- **Insert(subS, S, poz)** вставляет в строку S, начиная с позиции poz, подстроку subS;
- **Str(x, S)** преобразует число x в строковый формат;
- **Chr(n)** код символа; **Ord(ch)** символ.

Теоретический материал для C++

- `#include <iostream> ...` директивы
- `using namespace std ...` директива объявления пространства имен
- `const ...` описание констант
- Тип функции `<имя> (тип_аргумента аргумент_функции) ...`
функция
- {
- `setlocale(LC_ALL, "Russian");` // в Windows (где мы пишем текст программы) и в MS DOS (где отображается ее результат) используется разная кодировка русских букв. Эта команда служит для вывода русского текста.
- ...
- `system ("pause");` //остановка программы
- `return (возвращаемое значение);`
- }

Используемые директивы

- `#include <iostream>` // подключает в с++ функции, которые обеспечивают ВВОД/ВЫВОД.
- `#include <math.h>` //директива для использования математических функций
- `#include <iomanip>` //позволяет использовать манипуляторы
- `//пример cout<<setw(5)<<a` – означает, что на число a будет отведено ровно 5 ПОЗИЦИЙ
- `#include <time.h>` // для получения системного времени;
- `//пример: srand((type)time(time_delay))` – функция, позволяющая генерировать новые числа в рандоме по времени (type – тип генерируемых данных, time – генерация по времени, time_delay – задержка в миллисекундах);
`a=rand()%MAX_RANDOM±OFFSET` – генерация рандомных чисел (rand()% - сама функция, MAX_RANDOM – максимальное значение, OFFSET – смещение на числовой прямой значений)
- `using namespace std;` // эффект от ее применения состоит в том, что вы можете свободно применять команды ввода-вывода cout и cin. И процессор при этом будет четко понимать, что это команды, а не переменные.

Операции

- `X=5; //x присвоить пять`
- `cin>>x; // ввести x`
- `cin.getline(s,n)` – `s` – вводимая строка, `n` – количество символов
- `cout<<x; // вывести x`
- `cout<<endl; || cout<<"\n"; // вывести пустую строку (enter)`
- `cout<<"Hello, world! \n"; // вывести строку "Hello, world!" с переносом`
- `if (y) {д1}; else {д2}; //условие: Если условие (y) выполняется, то делаем действие (д1), иначе действие (д2)`
- `While (y) // пока условие в скобках истинно, выполняется тело подпрограммы`
- `for (парам=нач.знач; парам<кон.знач; парам++) тело цикла // для параметра цикла целочисленного типа с начальным значением до конечного значения (при < не включительно, при ≤-включительно) с шагом (шаг – увеличение счетчика на 1, для`

Операции

- Операции инкремента
- $i++$ постфиксная форма $i=i+1$;
- $++i$ префиксная форма $i=i+1$;
- Операции декремента
- $i--$ постфиксная форма $i=i-1$
- $--i$ префиксная форма $i=i-1$

Основные функции

- **strcpy(s1,s2)** – она копирует в строку *s1* содержимое строки *s2*;
- **strcmp(s1, s2)** – она сравнивает содержимое строк *s1* и *s2*: если они равны, то функция дает 0, если $s1 < s2$, то отрицательное число и если $s1 > s2$, то положительное;
- **itoa(l,s,r)** – преобразует целое *l* в строку *s*, пользуясь системой счисления с основанием *r*;
- **strlen(s)** – длина строки;
- **strstr(s1,s2)** - возвращает указатель на первое вхождение одной строки в другую

Примечание

- К сожалению, в C++ отсутствует тип «строка». Поэтому мы будем рассматривать строки как массивы СИМВОЛОВ.

Пример

- Условие:
 - Составить программу, которая определит, является ли заданное слово перевертышем.

Pascal

- **program** str;
- **var** s:string[10];
- l:integer;
- flag:boolean;
- **begin**
- flag:=true;
- readln(s);
- l:=length(s);
- **for var** i:=1 **to** l **do**
- **begin**
- **if** (s[i]<>s[l-i+1]) **then** flag:=false;
- **if** (flag=false) **then break**;
- **end**;
- **if** (flag=true) **then** writeln('Да, слово является перевертышем')
- **else** writeln('Нет, словно не является перевертышем');
- **end.**

C++

- `#include <iostream>`
- `using namespace std;`
- `int main()`
- `{`
- `char s[10];`
- `cin.getline(s,9);`
- `int l = strlen(s);`
- `bool flag = true;`
- `for (int i = 0; i < l; i++)`
- `{`
- `if (flag == false)`
- `break;`
- `}`
- `if (flag == true)`
- `cout << "Yes" << endl;`
- `else`
- `cout << "No" << endl;`
- `system("pause");`
- `return 0;`
- `}`

Задачи

- В символьную переменную вводится цифра. Вывести следующую и предыдущую цифры, считая, что за 9 следует 0, а, соответственно, нулю предшествует девятка;
- Вывести в одну строку нечетные (по порядковому номеру) буквы латинского алфавита: a c e g...;
- Дан текст, заканчивающийся точкой. Является ли этот текст правильной записью целого числа (возможно, со знаком);
- Дан текст, заканчивающийся точкой. Напечатать этот текст, удалив из него все цифры и знаки «+» или «-».