АЛГОхитрости

Типовые алгоритмические приёмы работы с символами и строками

1. Вывод на экран (печать) символа и его кода (номера в таблице ASCII).
2. Организация терминального условия в цикле до конца строки.
3. Символьные рамочные терминальные условия цикла.
4. Строка – целое число (в цикле).
5. Строка – вещественное число.
6. Удаление начальных пробелов в строке.
7. Особенности ввода строк оператором **scanf** со спецификатором формата ввода **%s**.

АЛГОхитрости

Типовые алгоритмические приёмы работы с символами и строками

1. **Вывод на экран (печать) изображения символа ASCII и его кода (порядкового номера в таблице).**

Pascal

Используются функции:

ord(F) - возвращает порядковый № символа хранящегося в перемен-ной F,

chr(№) - возвращает символ по его номеру.

Пример:

```
writeln('Символ: ', F,', Код: ', ord(F));
```

C / C++

Используются спецификаторы формата вывода:

Спецификатор формата **%c** выведет на экран собственно символ,

Спецификатор формата **%i (%d)** выведет на экран код символа.

Пример:

```
printf("Символ: %c Код: %i\n", F, F);
```

Примечание:

Операция `int n = 'h'+'e'+'l'+'l'+'o';` выполнит арифметическое суммирование значений кода символов: **n = 532**

АЛГОхитрости

Типовые алгоритмические приёмы работы с символами и строками

2. Организация терминального условия в цикле до конца строки.

Pascal

Используются функция:

Length (str) – текущая длина строки **str**.

Пример:

```
while (i <= Length(str));
```

C / C++

Терминальное условие автоматически становится ложным, когда встречает при пробеге по строке **0-символ** (\0, точнее \x0 – 16иричный ноль).

Пример:

```
while (str[i]); // цикл продолжается до \0
```

Но можно использовать функцию:

strlen (str) – текущая длина строки **str**,

Пример:

```
while (i <= strlen(str));
```

АЛГОхитрости

Типовые алгоритмические приёмы работы с символами и строками

3. Символьные рамочные терминальные условия цикла.

В качестве терминального условия в циклах и рекурсивных функциях обработки строк может использоваться рамочный диапазон символов:

Например,
для проверки попадания символа строки в диапазон символов-цифр терминальное условие при пробеге по строке будет:

```
while ((st[i] >= '0') && (st[i] <= '9'))
```
– в C

```
while (st[i]>='0') and (st[i]<='9') and (i<=Length(st))
```
– в Pascal

Pascal

АЛГОхитрости

C / C++

Типовые алгоритмические приёмы работы с символами и строками

4. Строка – целое число (в цикле).

```
Program StringInt;
(*Проверка  является ли строка целым числом*)
Var   st: string[20]; {строка текста}
        n:integer; {номер проверяемого символа}
begin
 writeln ('Введите число и нажмите
Enter');
    write('-> ');       readln(st);
 n:=1;
  while (n <= Length(st)) and
       ((st[n] >= '0')  and (st[n] <= '9'))
          do  n := n+1;
write('Введённая строка  ');
   if n < Length(st)
     then write ('не  ');
Writeln('является целым числом');
Writeln ('Для завершения нажмите
Enter');
Readln;
end.
```

```
#include <stdio.h>
#include <conio.h>
void main()
// Проверка, является ли строка целым числом
{
     char st[20]; // строка
      int i;       // номер проверяемого символа строки
  printf("\n Введите число и нажмите Enter
\n");
  printf ("-> ");
  scanf ("%s", st);
    i = 0;
    while ((st[i] >= '0') && (st[i] <= '9'))
      i++;
printf("/n Введённая строка ");
  if (st[i])     // st[i] -  \0, если введены только цифры
     printf("не ");
     printf("является целым числом.");
   printf("\n\nДля завершения нажмите
Enter");
   getch();
}
```

Pascal

АЛГОхитрости

Типовые алгоритмические приёмы работы с символами и строками

5а. Строка – вещественное число без знака.

```
Program StringReal;
(*Является ли строка дробным числом без знака*)
Var   st : string[20]; {строка текста}
        i  : integer; {номер проверяемого символа}
    err: boolean; {TRUE - строка не дробное число}
Begin  writeln('Введите число и нажмите Enter');
    write('-> ');
    readln(st);
    i:=1;
    err:=TRUE; {пусть строка - не дробное число}
if(st[i]>='1') and (st[i]<='9') {первый символ - цифра}
   then  begin {за цифрой могут быть ещё цифры}
```

см. продолжение

продолжение

```
While (st[i]>='0') and (st[i]<='9') and
(i<Length(st))          do  i := i+1;
{за цифрами следует точка, но она не последний символ}
  if (st[i] = '.') and (i < Length(st)) then
      begin
        i:=i+1; {за точкой должна быть хотя бы одна цифра}
        if (st[i]>='0') and (st[i]<='9')  then
        begin
         while (st[i]>='0') and (st[i]<='9') and
           (i<Length(st))   do i:=i+1;
          if i = Length(st)  {последний символ строки - цифра}
             then err:= FALSE; {строка – дробное число без знака}
  end; end; end;  write('Строка  ', st);
   if err      then write (' не  ');
Writeln(' является дробным числом без знака');
Writeln ('Для завершения нажмите Enter');
Readln;  end.
```

АЛГОхитрости

C / C++

Типовые алгоритмические приёмы работы с символами и строками

5б. Строка – вещественное число без знака.

```
#include <stdio.h>
#include <conio.h>
void main()
// Является ли строка дробным числом без знака
{  char st[20];   // строка
   int i;         // номер проверяемого символа строки
   int ok=0;   // пусть, строка - не дробное число
printf("\n Введите число и нажмите Enter\n");
  printf ("-> ");    scanf ("%s", st);
 i = 0;
if((st[i]>='0')&&(st[i]<='9')) //первый символ - цифра
   {  while ((st[i] >= '0') && (st[i] <= '9'))
         i++;
```

см. продолжение

продолжение

```
 if (st[i] == '.')  // за цифрами должна быть точка
        {  i++;
    // за точкой должна быть хотя бы одна цифра
      if ((st[i] >= '0') && (st[i] <= '9'))
        {   // и ещё цифры
          while ((st[i]>='0')&&(st[i]<='9'))
              i++;
            ok = 1;  // похоже строка - дробное число
        } } }
 printf("\nСтрока %s ", st);
   if (st[i] || !ok)
    printf("не ");
    printf("является дробным числом без знака.");
printf("\n\nДля завершения нажмите Enter");
  getch();   }
```

Pascal

АЛГОхитрости

C / C++

Типовые алгоритмические приёмы работы с символами и строками

6. Удаление начальных пробелов в строке.

```
program DELprobel;
{  удаляем из строки начальные пробелы }
Var
  str : string;
Begin
  writeLn('Введите строку: ');
  readLn(str);
// пока первый символ  пробел
 while ( str[1] = ' ' ) do
    delete(str, 1, 1);   // удаляем первый символ
  writeLn('Строка без начальных   пробелов:');
  writeLn(str);
end.
```

```
#include<stdio.h>
#include<string.h>
main()
{  int i;  char str[100]; // Массив для строки
printf("\nВведите строку символов:\n");
   gets(str);  // Вводим строку
   while (str[0]==' ') /*если первый символ
пробел, то сдвигаем массив влево  на одну позицию */
   {    for (i=0; i<strlen(str); i++)
str[i]=str[i+1];       }
printf("\nИсправленная строка:\n%s\n",str);
}
```

Более оптимально сделать так:

```
int k=0;
while (str[k]==' ')    k++;   // считаем пробелы
{      // смещаем строку  на k символов
    for (i=0; i<=strlen(str); i++)
        str[i]=str[i+k];    }
```

АЛГОхитрости

С

Типовые алгоритмические приёмы работы с символами и строками

7. Особенности ввода строк функцией scanf со спецификатором формата ввода %s.

Для чтения из входного потока строки можно использовать функцию scanf() со спецификатором преобразования %s. Использование спецификатора преобразования %s заставляет scanf() читать символы из буфера клавиатуры до тех пор, пока не встретится какой-либо разделитель. Читаемые символы помещаются в элементы символьного массива, на который указывает соответствующий аргумент, а после введенных символов еще добавляется символ конца строки ('\x0').

Разделителем может быть пробел, разделитель строк, табуляция, вертикальная табуляция или подача страницы. В отличие от функции gets(), которая читает строку, пока не будет нажата клавиша <ENTER>, scanf() читает строку до тех пор, пока не встретится первый разделитель. Это означает, что scanf() нельзя использовать для чтения строки "это испытание", потому что после пробела процесс чтения прекратится.

При вводе строки функцией scanf разделители, расположенные перед первым значащим символом строки – не вводятся (пропускаются).

Пример: ввод строки " привет всем":

```
#include <stdio.h>
int main(void)
{ char str[80];
printf("Введите строку: ");
scanf("%s", str);
printf("Вот Ваша строка: %s", str);
return 0; }
```

Программа выведет только часть строки, то есть слово "привет", без пробелов.