

# Счастливы́й билет

Разбор задания

# Вспомним условия задания

## Требования к программе

На вход программа принимает целое положительное число не превышающее 1 000 000. Если число является палиндромом, программа выводит сообщение YES, в противном случае NO. Если на вход была подана пустая строка, не числовое значение или отрицательное число, программа должна корректно завершиться, отобразив сообщение ERROR.

Ввод	Вывод	Пояснение
121	YES	Введённое число является палиндромом
10	NO	Введённое число не является палиндромом
7A	ERROR	На вход была подана строка, содержащая нецифровые символы
-10	ERROR	Программа не работает с отрицательными числами

# Дано

```
PROGRAM LuckyTicket;  
VAR  
    Num: INTEGER;  
    Error: BOOLEAN;  
  
//...  
  
BEGIN  
    Num := ReadNumber(Error);  
  
    WRITELN(Num);  
  
    IF Error  
    THEN  
        WRITELN('ERROR');  
END.
```

# Дано

```
PROGRAM LuckyTicket;
```

```
VAR
```

```
  Num: INTEGER;
```

```
  Error: BOOLEAN;
```

```
//...
```

```
BEGIN
```

```
  Num := ReadNumber(Error);
```

```
  WRITELN(Num);
```

```
  IF Error // Не хватает проверки на палиндром
```

```
  THEN
```

```
    WRITELN('ERROR');
```

```
END.
```

# Внесём правки

```
//...
```

```
BEGIN
```

```
  Num := ReadNumber(Error);
```

```
  IF Error
```

```
  THEN
```

```
    WRITELN('ERROR')
```

```
ELSE
```

```
  IF IsPalindrome(Num)
```

```
  THEN
```

```
    WRITELN('YES')
```

```
  ELSE
```

```
    WRITELN('NO')
```

```
END.
```

# Функция считывания числа

```
FUNCTION ReadNumber (VAR WasError: BOOLEAN): INTEGER;  
VAR  
    InputNum, Digit, ErrorCode: INTEGER;  
    Ch: CHAR;  
BEGIN  
    InputNum := 0;  
  
    WHILE NOT EOLN AND NOT WasError  
    DO  
        BEGIN  
            READ (Ch);  
            WasError := NOT IsDigit (Ch);  
            IF NOT WasError  
            THEN  
                BEGIN  
                    Val (Ch, Digit, ErrorCode);  
                    InputNum := Digit;  
                END  
            END;  
  
        RESULT := InputNum  
    END;
```

# Функция считывания числа

```
FUNCTION ReadNumber (VAR WasError: BOOLEAN): INTEGER;
VAR
    InputNum, Digit, ErrorCode: INTEGER;
    Ch: CHAR;
BEGIN
    InputNum := 0;
    // А если строка пустая?
    WHILE NOT EOLN AND NOT WasError
    DO
        BEGIN
            READ (Ch);
            WasError := NOT IsDigit (Ch);
            IF NOT WasError
            THEN
                BEGIN
                    Val (Ch, Digit, ErrorCode);
                    InputNum := Digit; // Присвоение последней цифры
                END
            END;
        END;

    RESULT := InputNum
END;
```

# Функция считывания числа

```
FUNCTION ReadNumber(VAR WasError: BOOLEAN): INTEGER;  
VAR  
    InputNum, Digit, ErrorCode: INTEGER;  
    Ch: CHAR;  
BEGIN  
    InputNum := 0;  
  
    IF EOLN  
    THEN  
        WasError := TRUE;  
  
    WHILE NOT EOLN AND NOT WasError  
    DO  
        BEGIN  
            READ(Ch);  
            WasError := NOT IsDigit(Ch);  
            IF NOT WasError  
            THEN  
                BEGIN  
                    Val(Ch, Digit, ErrorCode);  
                    InputNum := InputNum * 10 Digit;  
                END  
            END;  
  
        RESULT := InputNum  
    END;
```



# Проверка на палиндром

```
FUNCTION IsPalindrome(Number: INTEGER) : BOOLEAN;  
BEGIN  
    IsPalindrome := FALSE;  
END;
```

# Алгоритм проверки

1. Копируем число

12344321 => 12344321

2. Переворачиваем копию

12344321 => 12344321

3. Сравнить перевёрнутую копию с изначальным числом

12344321 = 12344321

# Реализация алгоритма

```
FUNCTION IsPalindrome(Number: INTEGER): BOOLEAN;  
VAR  
    ReversedNumber, OldNumber: INTEGER;  
BEGIN  
    OldNumber := Number;  
    ReversedNumber := 0;  
  
    REPEAT  
        ReversedNumber := ReversedNumber * 10 + Number MOD 10;  
        Number := Number DIV 10  
    UNTIL Number = 0;  
  
    IsPalindrome := OldNumber = ReversedNumber  
END;
```

Вопросы