

# Разбор задания №3

Система обнаружения

# Постановка задачи

На складе установлены датчики, ориентированные в одном из четырёх направлений (вверх, вниз, влево, вправо). Эти датчики испускают сигналы в выбранном направлении для сканирования коридоров. Сигнал продвигается по прямой, пока не встретит препятствие в виде стены или не выйдет за границы склада. Если на пути сигнала находится другой датчик, система фиксирует его и помечает как "обнаруженный". Задача — смоделировать прохождение сигнала от каждого датчика и определить, какие объекты будут зафиксированы системой.

Необходимо промоделировать испускание сигналов датчиками:

1. Если сигнал попадает в другой датчик, этот датчик помечается как обнаруженный (латинский символ 'O')
2. Если сигнал попадает в стену ('I' или '-') или выходит за границы склада, ничего не происходит
3. Сигналы в конечную точку прилетают мгновенно

# Постановка задачи

## Формат входных данных:

На вход программе подаются два числа  $N$  и  $M$  — длина и ширина склада.  $N$  и  $M$  находятся в диапазоне от 2 до 10. Затем вводится план склада размером  $N \times M$ , состоящее из символов:

- Стены:
  - 'I' (вертикальная стена)
  - '-' (горизонтальная стена)
- Пустое пространство:
  - '.'
- Датчики:
  - '>' (Датчик, ориентированный вправо)
  - '<' (Датчик, ориентированный влево)
  - '^' (Датчик, ориентированный вверх)
  - 'v' (Датчик, ориентированный вниз)

Гарантируется, что план склада вводится нужного размера.

# Постановка задачи

## Формат выходных данных:

- Если код выполнен успешно, то
  - На первой строчке пишется количество обнаруженных датчиков, если их нет, то написать **'NONE'**.
  - Показать склад после моделирования сигналов (с отображением обнаруженных датчиков).
- Если встречен неопознанный символ, вывести **'ERROR'** и завершить программу.
- Если N или M не входит в допустимый диапазон, вывести **'ERROR'** и завершить программу.

# Примеры входных и выходных данных

Ввод	Вывод	Объяснение
5 7 ..... .>>...< .-....-. .^...<. .....	3 2 3 2 7 4 2 ..... .>0...0 .-....-. .0...<. .....	Датчики с позиций (2;3) и (2;7) попали друг в друга. Датчик с позиции (2;2) попал в (2;3). Датчик с позиции (4;2) попал в стену. Датчик с позиции (6;6) попал в (4;2)
2 2 <> ..	NONE <> ..	Ни один из датчиков не попал в другие датчики
2 2 >! ..	ERROR	Неопознанный символ
52 ...	ERROR	Размер склада не должен превышать 10

# Заготовка

```
PROCEDURE ReadField(VAR M, N: INTEGER; VAR Field: FieldType);
VAR
    X, Y: INTEGER;
    Ch: CHAR;
BEGIN
    READLN(M, N);
    FOR Y := 1 TO M
    DO
        BEGIN
            FOR X := 1 TO N
            DO
                BEGIN
                    READ(Ch);
                    Field[X, Y] := Ch
                END;
            READLN
        END
    END;
END;
```

# Заготовка

```
PROCEDURE PrintField(Field: FieldType; M, N: INTEGER);  
VAR  
    X, Y: INTEGER;  
BEGIN  
    FOR Y := 1 TO M  
    DO  
        BEGIN  
            FOR X := 1 TO N  
            DO  
                WRITE(Field[X, Y]);  
            WRITELN  
        END  
    END  
END;
```

# Анализ заготовки

Что мы имеем:

- Процедура, считывающая поле
- Процедура, печатающая поле

Но:

- Нет обработки ошибок
- Не реализована работа датчиков



Можно сделать лучше

# Добавляем константы, типы и глобальные переменные

```
CONST
    MAX_SIZE = 10;
    MIN_SIZE = 2;
    SENSOR_SYMBOLS = ['^', 'v', '<', '>'];
    WALL_SYMBOLS = ['|', '-'];
    SPACE_SYMBOLS = ['.'];
    ALLOWED_SYMBOLS = SENSOR_SYMBOLS + WALL_SYMBOLS + SPACE_SYMBOLS;
    DETECTED_SENSOR = '0';
```

```
TYPE
    FieldType = ARRAY[1 .. MAX_SIZE, 1 .. MAX_SIZE] OF CHAR;
```

```
VAR
    InputField, DetectedField: FieldType;
    N, M, DetectedSensorsCount: INTEGER;
    Error: BOOLEAN;
```

# Добавляем обработку неправильной записи

```
PROCEDURE ReadField(VAR Error: BOOLEAN);
VAR
  X, Y: INTEGER;
  Ch: CHAR;
BEGIN
  Error := FALSE;
  READLN(M, N);
  IF NOT (N IN [MIN_SIZE .. MAX_SIZE]) OR NOT (M IN [MIN_SIZE .. MAX_SIZE])
  THEN
    BEGIN
      Error := TRUE;
      EXIT
    END;
  FOR Y := 1 TO M
  DO
    BEGIN
      FOR X := 1 TO N
      DO
        BEGIN
          READ(Ch);
          IF NOT (Ch IN ALLOWED_SYMBOLS)
          THEN
            BEGIN
              Error := TRUE;
              EXIT
            END;
          InputField[X, Y] := Ch;
          DetectedField[X, Y] := Ch
        END;
      READLN
    END
  END
END;
```

# Добавляем основной блок программы

```
BEGIN
  ReadField(Error);
  IF Error
  THEN
    BEGIN
      WRITELN('ERROR');
      EXIT
    END;
  SimulateAllEmissions;
  GetDetectedSensorsCount(DetectedSensorsCount);
  IF DetectedSensorsCount = 0
  THEN
    WRITELN('NONE')
  ELSE
    WRITELN(DetectedSensorsCount);
  PrintField(DetectedField)
END.
```

## Добавляем функцию симуляции работы датчиков

```
PROCEDURE SimulateAllEmissions;  
VAR  
    X, Y: INTEGER;  
BEGIN  
    FOR Y := 1 TO M  
    DO  
        FOR X := 1 TO N  
        DO  
            IF InputField[X, Y] IN SENSOR_SYMBOLS  
            THEN  
                SimulateEmission(X, Y);  
            END IF;  
        END DO;  
    END DO;  
END;
```

```

PROCEDURE SimulateEmission(StartX, StartY: INTEGER);
VAR
  X, Y: INTEGER;
  DeltaX, DeltaY: INTEGER;
BEGIN
  CASE InputField[StartX, StartY] OF
    '>': BEGIN DeltaY := 0; DeltaX := 1 END;
    '<': BEGIN DeltaY := 0; DeltaX := -1 END;
    '^': BEGIN DeltaY := -1; DeltaX := 0 END;
    'v': BEGIN DeltaY := 1; DeltaX := 0 END
  END;
  X := StartX + DeltaX;
  Y := StartY + DeltaY;
  WHILE (Y IN [1 .. M]) AND (X IN [1 .. N])
  DO
    BEGIN
      IF (InputField[X, Y] IN WALL_SYMBOLS)
      THEN
        EXIT;
      IF (InputField[X, Y] IN SENSOR_SYMBOLS)
      THEN
        BEGIN
          DetectedField[X, Y] := DETECTED_SENSOR;
          EXIT
        END;
      Y := Y + DeltaY;
      X := X + DeltaX
    END
  END;
END;

```

Добавляем функцию симуляции  
работы конкретного датчика

# Добавляем процедуру, подсчитывающую количество обнаруженных датчиков

```
PROCEDURE GetDetectedSensorsCount(VAR DetectedSensorsCount: INTEGER);  
VAR  
    Y, X: INTEGER;  
BEGIN  
    DetectedSensorsCount := 0;  
    FOR Y := 1 TO M  
    DO  
        FOR X := 1 TO N  
        DO  
            IF DetectedField[X, Y] = DETECTED_SENSOR  
            THEN  
                DetectedSensorsCount := DetectedSensorsCount + 1  
            END IF;  
        END DO;  
    END DO;  
END;
```

# Что изменилось

Теперь мы:

- Можем провести симуляцию работы датчиков, не затрагивая введенное поле
- Можем провести симуляцию работы конкретного датчика
- Обрабатываем ошибки



Вопросы