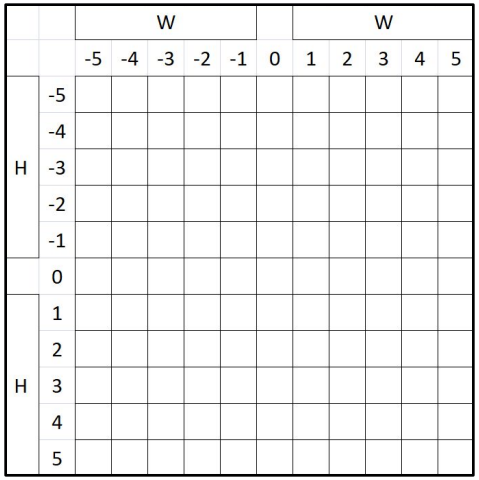
Разбор задания №3 (10-11кл)

Индиана Джонс

Постановка задачи

Индиана Джонс находится в центре поля на платформе с координатами (0; 0). Он спланировал маршрут, но столкнулся с проблемой: каждый раз, когда он шагает на платформу, предыдущая проваливается. При повторном посещении платформы Индиана падает в пропасть. Вам нужно определить, выживет ли Индиана, двигаясь по маршруту.



Постановка задачи

Формат входных данных:

Первая строка содержит три числа:

- W — количество платформ поля, располагающихся влево и вправо от центральной точки, принимает значения от 2 до 20,
- H — количество платформ поля, располагающихся вверх и вниз от центральной точки, принимает значения от 2 до 20,
- M — количество шагов, принимает значения от 2 до 1000.

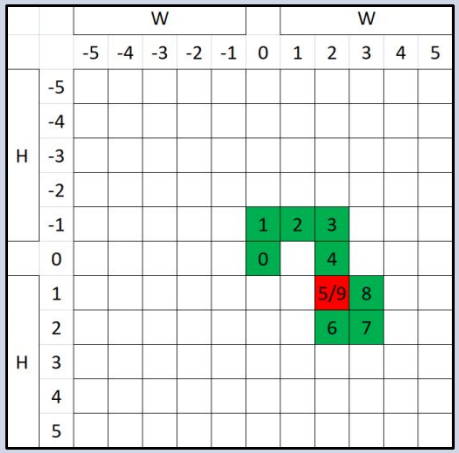
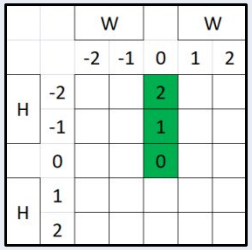
Затем следуют M строк, представляющих возможные шаги Индианы:

- 'U' - вверх,
- 'D' - вниз,
- 'L' - влево,
- 'R' - вправо.

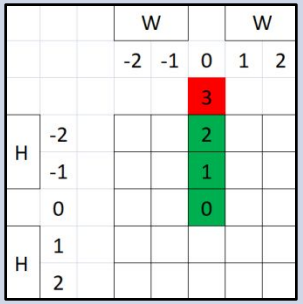
Формат выходных данных:

- ‘**SUCCESS’**, если Индиана выживет и не посетит одну и ту же клетку дважды.
- **'DIED in '** и координаты клетки (Y; X), если Индиана посетит одну и ту же клетку дважды.
- В случае неправильной команды, некорректных данных или выхода за пределы поля вывести **‘ERROR’** и завершить программу.

Примеры входных и выходных данных:

Входные данные	Выходные данные	Объяснение
5 5 9 U R R D D D R U L	DIED in 1 2	
2 2 2 U U	SUCCESS	

Примеры входных и выходных данных:

Входные данные	Выходные данные	Объяснение
2 2 1 U	ERROR	Неправильные входные данные
2 2 2 Up	ERROR	Неправильная команда
2 2 3 U U U	ERROR	Выход за пределы поля 

Простейшее решение

```
PROGRAM IndianaJonesAndTheLastCrusade(INPUT, OUTPUT);
CONST
  MaxHeight = 20;
  MaxWidth = 20;
  MaxNumberOfMoves = 1000;
  MinHeight = 2;
  MinWidth = 2;
  MinNumberOfMoves = 2;
  UpDirection = 'U';
  LeftDirection = 'L';
  DownDirection = 'D';
  RightDirection = 'R';
TYPE
  FieldType = ARRAY[-MaxHeight .. MaxHeight, -MaxWidth .. MaxWidth] OF BOOLEAN;
VAR
  Field: FieldType;
  Height, Width, NumberOfMoves: INTEGER;
  Error: BOOLEAN;
  //процедуры и основное тело программы
```

Простейшее решение

```
//переменные, процедуры, функции

BEGIN
  ReadParameters(Error);
  IF Error
  THEN
    WRITELN('ERROR')
  ELSE
    IndianaMoves
END.
```

Простейшее решение

```
PROCEDURE ReadParameters(VAR Error: BOOLEAN);
BEGIN
  Error := TRUE;
  READLN(Height, Width, NumberOfMoves);
  IF (Height < MinHeight) OR (Height > MaxHeight)
  THEN
    EXIT;
  IF (Width < MinWidth) OR (Width > MaxWidth)
  THEN
    EXIT;
  IF (NumberOfMoves < MinNumberOfMoves) OR (NumberOfMoves > MaxNumberOfMoves)
  THEN
    EXIT;
  Error := FALSE
END;
```

Простейшее решение

```
PROCEDURE IndianaMoves;
VAR
  Y, X: INTEGER;
BEGIN
  Y := 0;
  X := 0;
  Field[Y, X] := TRUE;
  WRITE('SUCCESS')
END;
```

Анализ решения

Решена часть задачи

- Ввод данных
- Вывод результатов
- Некоторые тесты пройдут

Не реализованы

- Обработка других ошибок
- Собственно прохождение маршрута Индианы

Хорошее решение

Улучшаем процедуру IndianaMoves

```
PROCEDURE IndianaMoves;
//инициализация переменных X, Y, MoveIndex
 FOR MoveIndex := 1 TO NumberOfMoves
  DO
    BEGIN
      ChangePos(Y, X, Error);
      IF Error
      THEN
        BEGIN
          WRITE('ERROR');
          EXIT
        END;
      IF Field[Y, X]
      THEN
        BEGIN
          WRITE('DIED in ', Y, ' ', X);
          EXIT
        END
      ELSE
        Field[Y, X] := TRUE
    END;
  WRITE('SUCCESS')
END;
```

Создаем функцию ChangePos

```
PROCEDURE ChangePos(VAR Y, X: INTEGER; VAR
      Error: BOOLEAN);
VAR
  Move: STRING;
BEGIN
  Error := FALSE;
  READLN(Move);
  IF LENGTH(Move) > 1
  THEN
     Error := TRUE
ELSE
  CASE Move[1] OF
    UpDirection:
     IF (Y - 1) >= -Height
     THEN
       BEGIN
         Y := Y - 1;
         EXIT
       END;
  //обработка других направлений (справа)
   END;
    Error := TRUE
END;
```

```
DownDirection:
   IF (Y + 1) <= Height
   THEN
     BEGIN
       Y := Y + 1;
       EXIT
     END;
  LeftDirection:
   IF (X - 1) >= -Width
   THEN
     BEGIN
       X := X - 1;
       EXIT
     END;
  RightDirection:
   IF (X + 1) <= Width
   THEN
     BEGIN
       X := X + 1;
       EXIT
     END
```

Что изменилось

Теперь у нас есть:

- Обработка ходов Индианы
- Обработка ошибок
- Чистый и понятный код