1. Wiersz: n, m, a, b

Co chce zrobić w programie:

Kolejka obiektow klasy miejsce, typu:

Poprzednie miejsce (int):

Następne miejsce (int):

Obiekty klasy droga, typu:

Miejsce początkowe(int):

Pod prąd(bool): (domyślnie false)

Miejsce końcowe(int):

Dzialanie programu:

Ile miejsc stworzyć – n;

Ile drog sprawdzić – m;

Start – a;

Koniec – b;

Odpowiedz: liczba złamanych praw (jazd pod prad)/-1 jeśli niemożliwe

1. Pobrane dane
2. Miejsca stworzone – przez symulacje, nie przez użytkownika
3. Pobieramy informacje o drogach (przepisach)
4. Próbujemy dotrzeć na miejsce

* Sortowanie obiektow droga:
  + Bierzemy pierwszy z brzegu,
  + Pobieramy dane – miejsce początkowe i miejsce końcowe,
    - Jeśli miejsce poczatkowe > miejsce końcowe -> pod prąd = true, zamieniamy miejscami wartość poczatkowa i koncowa,
    - Szukamy najmniejszej wartości miejsce poczatkowe i zapisujemy w tymczasowej zmiennej,
    - Jeśli przechodzi cale kolko bez zmiany wartości, to bierzemy to i jeśli to nie jest 1 to porównujemy z poprzednia zapisana wartoscia (aktywne miejsce),
    - Jeśli roznica miedzy nimi jest == +1 to idziemy dalej

Aktywne miejsce (wirtualne):

|  |  |
| --- | --- |
| Poprzednie miejsce | 0 |
| Następne miejsce | 1 |

Stos:

Pierwsza w kol. Wprow. droga

|  |  |
| --- | --- |
| Miejsce początkowe | 3 |
| Miejsce końcowe | 2 |
| Pod prąd | Nie |

druga w kol. Wprow. droga

|  |  |
| --- | --- |
| Miejsce początkowe | 1 |
| Miejsce końcowe | 2 |
| Pod prąd | Nie |

trzecia w kol. Wprow. droga

|  |  |
| --- | --- |
| Miejsce początkowe | 4 |
| Miejsce końcowe | 3 |
| Pod prąd | Nie |

czwarta w kol. Wprow. droga

|  |  |
| --- | --- |
| Miejsce początkowe | 4 |
| Miejsce końcowe | 5 |
| Pod prąd | nie |

1. **Szukanie drog pod prad i zamienianie kolejności polaczen**

Stos:

Pierwsza w kol. Wprow. droga

|  |  |
| --- | --- |
| Miejsce początkowe | **2** |
| Miejsce końcowe | **3** |
| Pod prąd | **tak** |

druga w kol. Wprow. droga

|  |  |
| --- | --- |
| Miejsce początkowe | 1 |
| Miejsce końcowe | 2 |
| Pod prąd | Nie |

trzecia w kol. Wprow. droga

|  |  |
| --- | --- |
| Miejsce początkowe | **3** |
| Miejsce końcowe | **4** |
| Pod prąd | **tak** |

czwarta w kol. Wprow. droga

|  |  |
| --- | --- |
| Miejsce początkowe | 4 |
| Miejsce końcowe | 5 |
| Pod prąd | nie |

**No i tu wystarczy:**

1. **Szukac najmniejszej wartości miejsca początkowego**
2. **Sprawdzić czy zgodna z poprzednia koncowa**
3. **Jak tak to Sprawdzić czy jest polaczenie do kolejnej wg ciagu liczb naturalnych**
4. **Jak tak to Ustawić jako droga aktywna**
5. **Sprawdzić czy pod prad – jak tak to zwiekszyc licznik**

**Powtórzenie petli:**

1. **Szukac najmniejszej wartości miejsca początkowego**
2. **Sprawdzić czy zgodna z poprzednia koncowa**
3. **Jak tak to Sprawdzić czy jest polaczenie do kolejnej wg ciągu liczb naturalnych**
4. **Jak tak to ustawić jako droga aktywna**
5. **Sprawdzić czy pod prad – jak tak to zwiekszyc licznik**

**Bledy petli:**

1. **– punkt 2) – jesli nie – oznaczyć jako niepotrzebna/usunąć – szukac nowej wg petli**
2. **– punkt 3) – jeśli nie to sprawdzić czy jest pod prad – jeśli nie to sprawdzić czy jest mniejsza od docelowej i skorzystać o ile możliwe, następnie przyjąć jako aktywna i wrocic do petli lub jeśli niemożliwe – oaznaczyc jako niepotrzebna/usunąć lub jeśli jest pod prad to odstawić i szukac innej, jeśli brak – uzyc tej**
3. **– punkt 5) jak nie to przejechać i zacząć petle od nowa**

**Test:**

1. **1-2**
2. **Tak**
3. **Tak**
4. **Droga aktywna**
5. **Nie**
6. **2-3**