## Labyrint 4

Tento program je simulační program pro roznáškového robota, který roznáší zásilky v 2D labyrintu.

Chceme připravit plánovací program, který mu zjistí, za jak dlouho dokáže danou zásilku převézt z bodu A do bodu B.

Robot dokáže přepravit zásilku o jeden metr (nahoru, doprava, dolů a doleva) za jednu (1) vteřinu na normální cestě (.). Pokud robot vjede do mlhy (f), trvá mu čtyři (4) vteřiny z mlhy vyjet.

Robot nikdy nesmí narazit (vjet) do zdi (#).

Robot vždy vyráží z cesty (.) a doručuje zásilku na cestu (.).

Náš program by pro robota měl zjistit, za jak **nejkratší dobu** (kolik nejméně **vteřin**) mu zabere přenesení zásilky z bodu A do bodu B.

Robot vždy vyráží z bodu A

robot je již na místě A a spouští si náš program, aby zjistil, jak dlouho bude jezdit v labyrintu

### Popis vstupu (část první)

Program dostává na standardním vstupu definici labyrintu

Vstup programu lze rozdělit do dvou fází

- plánek labyrintu
- oddělovač pomlčka -
- sekvence otázek na výpočet nejkratšího času z bodu A do bodu B

Plánek labyrintu vypadá následovně

- . (tečka) označuje cestu, po které lze projít
- f označuje mlhu, ze které se obtížně vyjíždí
- # (hash) označuje zeď, přes kterou nelze projít
- labyrint nemusí být ohraničen zdmi
- každý "řádek" labyrintu je zakončen znakem nové řádky (\n)
- každá sousední dvojice znaků (v řádku) je oddělena třemi mezerami

Pokud robot přijel k samotnému okraji labyrintu a ve směru jeho jízdy již plánek nepokračuje, robot může pokračovat na políčko druhé strany labyrintu. Stále ale platí pravidlo, že dané políčko nesmí být zeď.

Podobně jako Pac-Man.

```
# # # #
# . # #
# . . #
# # # #
```

Následuje oddělovač fází vstupu - pomlčka (-) a znak nové řádky (\n).

Tedy:

```
# # # #
# . # #
# . . #
# # # #
```

Poté následuje druhá fáze vstupu.

### Popis vstupu (část druhá)

Druhá fáze vstupu obsahuje sekvenci startovních (A) a cílových bodů (B), na které program odpovídá (na standardní výstup) číselný údaj, reprezentující minimální počet vteřin, který bude muset robot strávit přesunem dané zásilky nebo -1, pokud danou zásilku nemůže doručit (cesta neexistuje).

Odjezd z políčka typu cesta (.) trvá jednu (1) vteřinu.

Odjezd z políčka typu mlha (f) trvá čtyři (4) vteřiny.

Startovní i cílový bod bude vždy umístěn na cestě (.).

Body A, B jsou zadány jako čtveřice čísel (celých, nezáporných) zakončených novou řádkou ( $\n$ )

Tato čtveřice reprezentuje souřadnice v labyrintu.

Formát lze popsat takto: <index řádku startovnáho bodu> <index sloupce startovního bodu> <index řádku koncového bodu> <index sloupce koncového bodu>

Indexy jsou počítany od 0 a zvyšují se směrem doprava a dolů. Bod 0 0 je políčko v levém horním rohu.

Může se to zdát divné, ale robot může dostat zakázku na doručení zásilky na stejné místo, ze kterého ji má vyzvednout, v tomto případě je počet vteřin strávený přesunem zásilky roven 0.

# Ukázky vstup a výstupu

### Ukázka 1

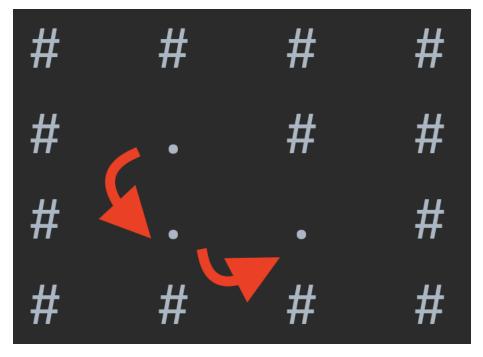
## Vstup:

# # # # # # . # # # . . # # # # # -1 1 2 2 2 2 1 1 2 1 2 2

## Výstup:

2 2 2 1

Vysvětlení prvního zadání (1 1 2 2 ==> 2)



#### Ukázka 2

```
Vstup:
```

\_

- 1 1 3 5
- 2 3 3 5
- 3 1 3 5
- 3 5 2 1

## Výstup:

- 10
- 5
- 8
- 9

#### Ukázka 3

#### Vstup:

-

- 1 1 3 5
- 2 1 3 5
- 3 1 3 5
- 3 2 3 5

## Výstup:

- 6
- 7
- 8
- 7

#### Ukázka 4

```
Vstup:
```

```
#
   f
#
   #
```

1 1 1 3

1 3 1 1

Výstup:

5

5

#### Ukázka 5

### Vstup:

```
f
#
```

1 1 1 3

1 3 1 1

Výstup:

4

4

## Ukázka 6

### Vstup:

```
f
  #
#
```

1 1 1 3

1 3 1 1

Výstup:

5 5

#### Ukázka 7

```
Vstup:
```

-

0 1 3 3 3 3 0 1

#### Ukázka 8

#### Vstup:

1 1 1 3

Výstup:

5

### Ukázka 9

#### Vstup:

1 1 3 5

2 3 3 5

3 1 3 5

3 5 2 1

# Výstup:

8

5

8

7