

Projekt 2 – Tézus a Minotaurus

*„Za ohybom chodby uvidel Tézus Minotaura.
Stál nad hromadou vybielených kostí ...“*

Tézus bol veľmi odvážny mladík. Zauzomienil si, že zneškodní Minotaura, vraj obludu, vraj napol býka napol človeka.

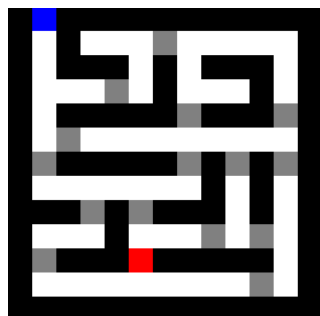


K tomu si zaobstaral mapu labyrintu, v ktorej sú steny, chodby a význačné body – dvere. Mapa je zadaná ako dvojrozmerná matica znakov: stena je vyznačená znakom mriežka (#), chodba znakom bodka (.), a význačný bod veľkým písmenom anglickej abecedy (každé písmeno je použité najviac raz). Tézus začína vpravo hore v labyrinte vo význačnom bode T. Predpokladané miesto výskytu Minotaura je vo význačnom bode X.

Ukážka takého labyrintu:

```
#T#####  
#.#...R....#  
#.#...#.#...#  
#...Q.#...#.#  
#.#...C###F#  
#.#A.....#  
#B#####E#K#L#  
#.....#.#.#  
###D#H###.#.#  
#...#...J.P.#  
#G###X#####.#  
#.....N.#  
#####
```

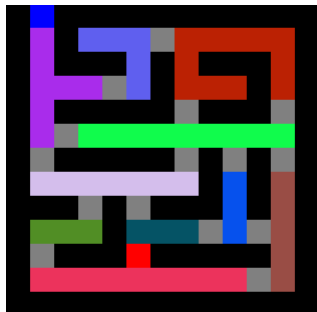
Prvou úlohou je vykreslenie políček labyrintu do BMP obrázka podľa ukážky nižšie. Nakreslite do súboru obrázkov, v ktorom sú políčka labyrintu zafarbené nasledovne: steny sú čierne, chodby sú biele, význačné body sú sivé, Tézus je modrý a Minotaurus je červený.



Súvislé oblasti chodieb bez dverí – tzv. miestnosti – tvoria určitý taktický prvok v boji proti Minotauru. Tézus chce v labyrinte postupovať po miestnostiach, potrebuje teda do obrázku vyznačiť políčka jednotlivých miestností rovnakou farbou. Konkrétna farba nie je dôležitá ale

políčka labyrintu prislúchajúce do rôznych miestností by mali mať odlišnú farbu. Políčko s Minotaurom (X) považujte pre túto úlohu ako dvere.

Druhou úlohou je ofarbenie políčok miestností labyrintu do BMP obrázka podľa ukážky nižšie. Nasleduje jedna možná ukážka zafarbenia labyrintu:



Využitím tejto zafarbenej mapy miestností chce Tézus analyzovať následnosť medzi význačnými bodmi, teda ktoré význačné body sú susedné: z ktorého význačného bodu sa dá dostať do ktorých ostatných význačných bodov bez prechodu iným význačným bodom.

Napr. z význačného bodu T sa dá dostať do bodov A, B a Q. Do iných sa nedá dostať bez toho aby prešiel cez niektoré z A, B alebo Q.

Tretou úlohou je nájsť a vypísať na štandardný výstup zoznamy susedných význačných bodov. Pre labyrint vyššie to sú tieto:

A: L K F E C T Q B
B: H E D T Q A
C: L K F E A R F
D: G H E B
E: H D B L K F C A
F: L K E C A R C
G: X N D
H: X J E D B
J: X H P K
K: P J L F E C A
L: P N K F E C A
N: X G P L
P: N L K J
Q: R T B A
R: F C Q
T: Q B A
X: N G J H

Čiže napr. z bodu H sa dá priamo dostať do X, J, E, D a B ale nie do iných. Jednoduchá reprezentácia relácie susednosti v tejto úlohe sa dá realizovať aj dvojrozmernou maticou 26x26, vašou úlohou je vytvoriť pre každý význačný bod spájaný zoznam význačných bodov, do ktorých sa dá priamo dostať.

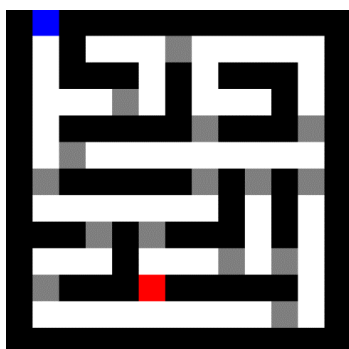
Vyzbrojený týmito informáciami sa nebojácne Tézus vypravil hľadať a bojovať s Minotaurom...

Minotaurus vie, že mu idú po krku, napriek tomu si chce pred bojom dobre pospať. Predpokladá, že čím hlbšie v labyrinte bude oddychovať, tým je menšia pravdepodobnosť, že ho bojovníci nájdu počas spánku. Každým prechodom dverí (význačný bod) sa zvyšuje hĺbka postupu bojovníkov a Minotaurus by chcel v mape vyznačiť kde všade sa bojovníci (Tézeus) môžu v labyrinte dostať ak budú odvážni prejsť až K dverí (význačných bodov).

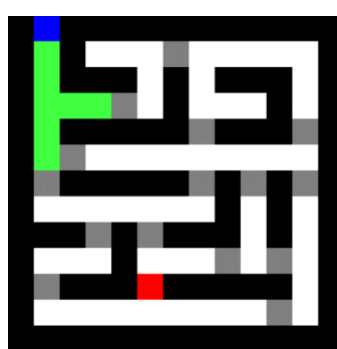
Štvrtou úlohou je pre rôzne hodnoty K vyznačiť v labyrinte políčka miestností do ktorých sa Tézeus môže dostať ak prejde najviac K dverí. Vstup do labyrintu políčko (T) považujte pre túto úlohu ako dvere.

Napr. pre hodnoty K=0, 1 a 2 sú políčka (zafarbené na zeleno) nasledovné:

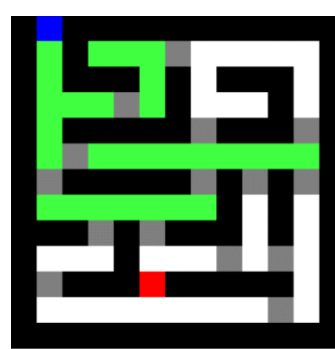
K=0



K=1



K=2



Kreslenie labyrintu do BMP súborov môžete implementovať do týchto funkcií:

- `void kresli_mapu(char *meno_suboru, int k):` ktorá pre hodnotu K vykreslí príslušnú mapu labyrintu. Mapa pre K=0 je riešenie prvej podúlohy, pre K>0 je riešením štvrtej podúlohy.
- `void kresli_miestnosti(char *meno_suboru):` ktorá vykreslí zafarbené políčka jednotlivých miestností labyrintu. Toto je riešenie druhej podúlohy.

Farby môžete zvoliť iné, ale musia byť také, aby boli ľahko rozlíšiteľné.

Program by mal byť napísaný v programovacom jazyku C, zachováva konvencie písania prehľadných programov. Snažte sa, aby to bolo na prvý pohľad pochopiteľné.

Konzultácie k riešenia projektu: do 11. týždňa semestra.

Termín odovzdania programu do testovača: 11. týždeň do začiatku vlastného cvičenia.

Prezentácia programu, riešenia projektu: 10. - 12. týždeň semestra.

Prajeme veľa úspechov!