

# Tema de casă 1 - DwarfLand

---

## Responsabili:

- Claudia Ifrim (2016) [mailto:ifrim.claudia@gmail.com]
- Teodora Serbanescu (2016) [mailto:teodora.serbanescu@stud.acs.upb.ro]

**Deadline:** 06.11.2016

**Ultima modificare:** 23.10.2016

## Updates:

- 26.10.2016: checker.zip a fost modificat

Menționăm că pentru testare (pe vmchecker) se folosește o mașină virtuală pe 32 de biți. Arhiva de test folosește un astfel de binar. În caz că sistemul vostru de operare de pe mașina fizică este pe 64 de biți, sugerăm să faceți testarea finală și pe o mașină (virtuală sau nu) de 32 de biți.

## Obiective

---

În urma realizării acestei teme, studentul va fi capabil:

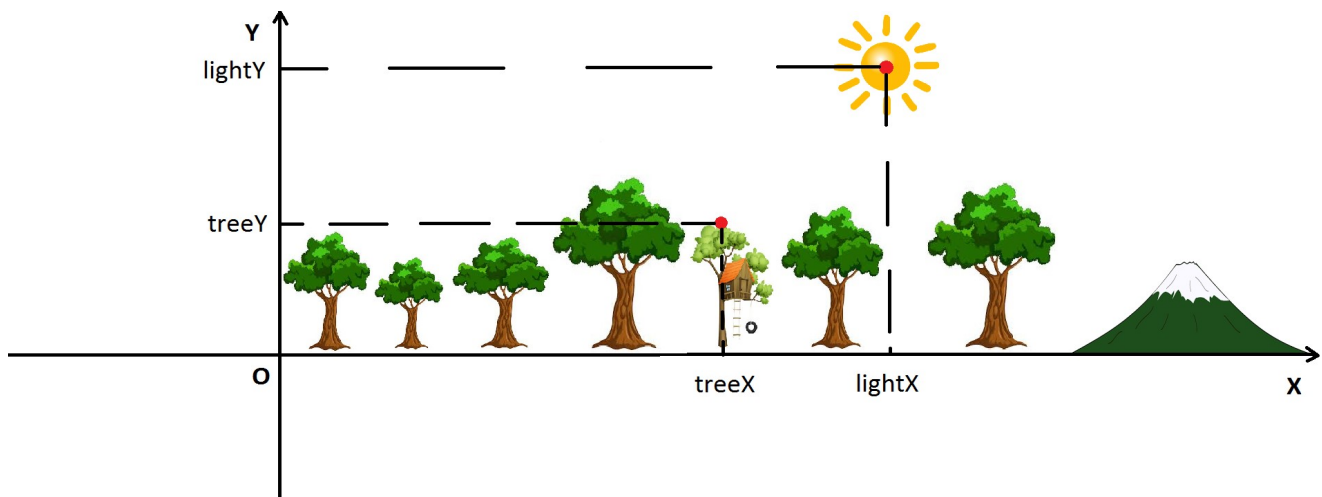
- să utilizeze funcțiile din limbajul C de citire și scriere a datelor
- să utilizeze funcțiile matematice din C
- să utilizeze structuri condiționale și de repetiție din C
- să utilizeze programul Make pentru a compila automat programele scrise

## Enunțul temei

---

Pentru a spori siguranța lor, o familie de pitici din DwarfLand a decis că cea mai sigură soluție pentru construcția casei lor în care vor locui pe timp de iarnă este în unul dintre copacii ce sunt aliniați de-a lungul drumului drept ce duce către muntele Magic. Totuși, pentru confortul lor, ei vor ca locuința lor să aibă parte de cât mai multă lumină.

Pentru a putea începe construcția casei cât mai repede, unul dintre pitici a descoperit un algoritm care generează soluția optimă pentru alegerea copacului în care își vor construi casa. Algoritmul său generează o soluție luând în considerare faptul că arborii au diferite înălțimi și sunt perfect aliniați de-a lungul drumului și cunoscând punctul (lightX, lightY) în care se află unica sursă de lumină și intensitatea acesteia (lightSourceIntensity).



Conform formulei Inverse Square Law [[https://en.wikipedia.org/wiki/Inverse-square\\_law](https://en.wikipedia.org/wiki/Inverse-square_law)], intensitatea luminii într-un punct este:

$$I(p) = 1/\text{dist}(p,p_0)^2 * I(p_0)$$

unde:

- $p$  = este punctul în care calculăm intensitatea;
- $p_0$  = este punctul în care se află sursa de lumină;

Copacul ideal pentru poziționarea locuinței trebuie să respecte următoarele condiții:

- numărul format din ultima cifră a intensității luminii în vârful său (partea întreagă a acesteia), urmată de ultimele cifre ale înălțimii copacului precedent, curent și următor este număr magic.
  - Exemplu:
    - $I = 77.6 \Rightarrow 77$
    - $h_1 = 373$
    - $h_2 = 753$
    - $h_3 = 991$
  - numărul cerut este: 7331, iar acesta este număr magic dacă:
    - 7 este număr prim
    - 73 este număr prim
    - 733 este număr prim
    - 7331 este număr prim

Primul și ultimul copac nu sunt considerați arbori ideali pentru poziționarea unei locuințe deoarece sunt expuși pericolelor.

Dacă există mai mulți copaci ideali pentru construirea casei, se va alege cel cu cea mai mică înălțime, deoarece, nu uitati, este țara piticilor. Dacă doi sau mai mulți copaci ideali au aceeași înălțime(cea mai mică), se va afișa cel mai din stânga(cu coordonata X minimă).

În cazul în care nu există niciun astfel de arbore, se va afișa textul "There is no place for you in DwarfLand!".

## BONUS

Piticii consideră că ar putea construi și o grădină suspendată după trecerea iernii în cazul în care copacul ales permite această extindere. În cazul în care există un copac ideal, să se calculeze și unghiul format de varful său (acesta fiind considerat ca fiind vârful unghiului) și vârfurile copacilor vecini. Piticii consideră că un copac permite construirea unei grădini suspendate în cazul în care unghiul format de acesta cu vecinii săi este obtuz.

Funcțiile trigonometrice din C, precum *asin*, *acos*, *atan*, *atan2* returnează măsurile unghiurilor măsurate în radiani. Pentru a face conversia în grade, veți avea nevoie de o constantă **PI**, pe care vă recomandăm să o definiți ca: **3.14159265**.

## Cerința temei

Scrieți un program C pentru a valida corectitudinea algoritmului descoperit de către pitic.

**BONUS:** În cazul în care se găsește un copac ideal, să se calculeze și unghiul format de varful său (acesta fiind considerat ca fiind vârful unghiului) și vârfurile copacilor vecini. Unghiul va fi măsurat în sens trigonometric.

### Date de intrare:

- `lightX lightY lightSourceIntensity`
- $n$  - numărul de copaci ce sunt aliniați de-a lungul drumului
- $n$  linii cu coordonate  $(x,y)$  ale copacilor
- **Observație:** Coordonatele  $x$  ale copacilor sunt date în ordine crescătoare.

### Date de iesire:

- Indicele copacului ideal în care piticii își vor construi casa, dacă acesta există, (copacii sunt numerotați începând cu 0) urmat de intensitatea luminii calculată în vârful acestuia. (Afișarea acestei valori se va face folosind o precizie de 4 zecimale)
- **BONUS:**

- valoarea unghiului măsurat (Afișarea acestei valori se va face folosind o precizie de 4 zecimale)
- se va afișa mesajul **"IT CAN BE EXTENDED"** dacă unghiul este obtuz sau mesajul **"IT CAN NOT BE EXTENDED"** în caz contrar.

## Exemplu

---

### Input

```
15 23 7760.00
6
2 10
7 11
10 13
15 13
20 11
22 14
```

### Output

```
3 77.6000
158.1986 IT CAN BE EXTENDED
```

## Precizări legate de implementarea temei

---

- $treeXi < treeXi+1, 0 \leq i < n - 1$
- $2 < n < 100$
- toate coordonatele din plan sunt numere întregi din intervalul  $[0, 1000]$
- `lightSourceIntensity` este un număr real mai mic decât  $2^{24}$
- Unghiul măsurat are o valoare între 0 și 180 de grade
- În implementarea temei NU este permisă folosirea vectorilor
- Afișarea intensității luminii și a unghiului măsurat se va face folosind o precizie de 4 zecimale
- Deși programul vostru va trebui să citească direct de la tastatură și să afișeze pe ecran (folosind, de exemplu, `scanf` și `printf`), puteți să citiți datele și să le scrieți în fișiere, folosind redirectările din consolă, fără să modificați programul. Pentru mediul Windows, dacă fișierul de intrare este `dwarfland.in`, și cel de ieșire este `dwarfland.out`, iar programul vostru se numește `dwarfland.exe`, veți tipări la consolă:

```
dwarfland.exe < dwarfland.in > dwarfland.out
```

- Pentru Linux, comanda este similară:

```
./dwarfland < dwarfland.in > dwarfland.out
```

## Barem de corectare

---

Punctajul va fi împărțit astfel:

- **50p** - afișarea indicelui corect
- **30p** - afișarea corectă a intensității luminii
- **20p** - explicațiile din README și aspectul codului sursă (vezi coding style [<http://ocw.cs.pub.ro/courses/programare/coding-style>])
- **10p** - afișarea valorilor corecte pentru bonus

## Testare

---

- Arhiva de testare folosită pe `vmchecker` va fi următoarea: `checker.zip` [[http://ocw.cs.pub.ro/courses/\\_media/programare/teme\\_2016/checker.zip](http://ocw.cs.pub.ro/courses/_media/programare/teme_2016/checker.zip)]

- În arhiva checker.zip aveți un script pentru testarea temei - implementarea fără bonus - (./checker.sh) și un script (./checker\_bonus.sh) pentru testarea temelor ce au implementat și bonusul.
- checkerul și directoarele cu inputuri și outputuri vor fi dezarhivate în același director în care se află implementată tema, precum și fisierul Makefile

Copierea parțială sau totală a unei rezolvări din altă sursă va atrage după sine anularea punctajelor pentru toate temele de casă, atât pentru cel care a copiat, cât și pentru sursa acestuia.

## Regulament teme

---

- Tema se va implementa DOAR în limbajul C. Va fi compilată și testată DOAR într-un mediu LINUX. Nerespectarea acestor reguli aduce un punctaj nul pe temă.
- Tema va fi trimisă atât pe vmchecker [<https://elf.cs.pub.ro/vmchecker/>] cât și pe moodle [<http://cs.curs.pub.ro>], sub forma unei arhive ZIP. Nerespectarea acestei reguli aduce un punctaj nul pe temă.
- Fișierele temei trebuie OBLIGATORIU împachetate într-o arhivă de tip '.zip', cu numele 'Grupa\_NumePrenume\_Tema1.zip'.
- Arhiva va trebui să conțină în directorul RADACINA doar următoarele:
  - 1. Codul sursă al programului vostru (fișierele .c și eventual .h).
  - 2. Un fișier Makefile care să conțină regulile build și clean. Regula build va compila programul într-un executabil cu numele **dwarfland**. Regula clean va șterge executabilul și eventual toate binarele intermediare (fișiere obiect) generate de voi.
  - 3. Un fișier README care să conțină prezentarea implementării alese de voi. Dacă ați implementat și bonusul, menționați acest lucru în README.
  - Nerespectarea regulilor 1 și 2 aduce un punctaj nul pe temă.
- Arhiva temei NU va conține fișiere binare.
- O temă care nu compilează pe vmchecker [<https://elf.cs.pub.ro/vmchecker/>] va fi punctată.
- O temă care compilează dar care NU trece niciun test pe vmchecker [<https://elf.cs.pub.ro/vmchecker/>] nu va fi punctată. Punctele pe teste sunt cele de pe vmchecker.