

# Artificial Intelligence Project

Noor Khamaisi – 207076076

Shadi Halloun - 313552309

## הקדמה:

קודם כל אנחנו מתחילים בזה שהפרייקט היה מעניין מאוד וכל כך תרם לנו ידע באלגוריתמים כל כך חשובים וניסיון תכנותי שיכול לעזור לנו גם לעתיד.

העבודה על הפרוייקט לא הייתה קלה, ומיוחד בתקופה הזו של הסמסטר שהייתה די עמוסה, מה שהקשה את המשימה עבורינו כי היה לנו מבחנים ותרילי בית שהם חובה להגשה לכן נאלצנו להתחיל בתרגילי חובה קודם ורק אז לעבור לפרוייקט, כשהגענו לפרוייקט, הגענו בתקופה מאוחרת של הסמסטר ולא היה לנו מספיק זמן לעבוד על הפרוייקט, אבל בכל זאת, כל כך רצינו לעשות אותו כי ידענו שזה נותן לנו את ההזדמנות להכיר אלגוריתמים חשובים בבינה מלאכותית, עבודה משותפת, ניסיון תכנותי, הזדמנות לעבוד ב-GitHub וגם עוזר לנו לציון הסופי, לכן דרשנו מהמרצה עוד הארכה כדי שנוכל לנצל ההזדמנות הזו ואנחנו מודים למרצה על הבנתו והסבלנות כלפינו.

היה פרוייקט מאתגר ולא היה קל לממש אלגוריתמים כאלה כי זה דורש הבנה עמוקה לאלגוריתמים קודם כל, ורק אז להתחיל בכתיבת הקוד.

## קצת הסברים על ההיוריסטקות:

עשינו מלא חיפוש כדי לדעת איזה היוריסטקות טובות וכדאי לנו לעבוד איתם, ולבסוף החלטנו לבחור בהיוריסטקה אוקלידית והיוריסטקת מנהטן.

### 1- ההיוריסטקה האוקלידית:

היא היוריסטקה שמחשבת את המרחק האוקלידי מנקודה מסויימת עד הסוף, זאת היוריסטקה אדמיסבילית לכן היא עובדת טוב ונותנת לנו פתרון אופטימלי.  
**הוכחת אדמיסביליות :** עבור כל צומת בעץ היא נותנת לנו חסם תחתון לעלות האמיתית עד לצומת היעד שלנו כלומר היא אף פעם לא מעריכה יותר מהעלות בפועל, וזה נובע מכך שהמרחק האוקלידי עצמו עד נקודה מסויימת יותר קטן מהמרחק האמיתי בפועל.

### 2- היוריסטקת מנהטן:

היא היוריסטקה שמחשבת את מרחק מנהטן מנקודה מסויימת עד הסוף.  
בהתחלה בחרנו לעבוד עם היוריסטקה זאת אבל אחר כך שמנו לב שהיא לא טובה לבעיות אלה שיש לנו שניתן לזוז בהם גם באלכסון, לכן נאלצנו לעשות שינוי להיוריסטקה על מנת להפוך אותה לאדמיסביליות ותניב לנו תוצאות טובות.  
השינוי שעשינו הוא שחלקנו את תוצאת מרחק מנהטן בגודל הקפיצה הגדולה ביותר עד היעד שלנו, וזה נותן לנו אדמיסביליות כי כשמחלקים גודל הקפיצה ביותר עד היעד, אנחנו תמיד נקבל עלות שהיא קטנה יותר מהעלות בפועל.

- בסוף בחרנו לעבוד עם ההיוריסטקה האוקלידית, כי ראינו שהיא נותנת לנו תוצאות יותר טובות. (זמן, מספר הצמתים שנפרסו/נסרקו, עומק החיפוש ...)

## קצת הסברים על האלגוריתמים:

### -1 A\* Algorithm :

אלגוריתם ראשון שהתחלנו בו, היה יותר קל יחסית למימוש משאר האלגוריתמים, עשינו אינסוף חיפושים בגוגל ובהרצאות על מנת להבין אותו לעומק ולדעת איך להתחיל. ניסינו את שתי ההיוריסטקות שבחרנו עם האלגוריתם הזה אבל ראינו שההיוריסטקה האוקלידית נותנת לנו תוצאות יותר טובות. בהתחלה מימשנו את האלגוריתם הזה עם Open List שהיה מערך, ובכל איטרציה היינו ממיינים את המערך לפי ה  $F \text{ cost}$  ופורשים את האיבר הראשון. אחר כך, שמנו לב שאפשר לשפר את זמן הריצה של האלגוריתם ולכן החלפנו את המערך בערימת מינמום ובאמת קיבלנו זמן ריצה יותר טוב.

### -2 UCS Algorithm :

אלגוריתם שני שעברנו אליו – אלגוריתם היה קל יחסית והצלחנו לפתור אותו במהירות – כי הוא אותו מימוש כמו אלגוריתם ASTAR רק שכאן ההיוריסטיקה היא קבועה (אפס).

### -3 IDA\* Algorithm :

אלגוריתם זה הוא האלגוריתם הקשה ביותר – הוא זה שלקח לנו הרבה זמן עבודה עד שהצלחנו לפתור אותו בצורה הטובה ביותר, תאמת שהצלחנו לקבל תוצאה כבר מההרצה הראשונה, אבל הוא היה לוקח זמן גדול עד שנקבל את הפתרון – על המטריצה הקטנה  $(16*16)$  היה לוקח 3 דקות ועל  $(30*30)$  היה לוקח יותר מחצי שעה. כל זה היה בגלל שהמימוש שלנו היה באמצעות רקורסיה – כלומר עשינו אותו מימוש כמו זה שנמצא במצגות, אבל הוא היה לוקח זמן רב כי כל צומת הוא פורש הרבה פעמים – כלומר כל פעם הוא לוקח את השכנים ובשכנים אלה יש צמתים שהם כבר נלקחו ונפרשו בעבר. עבדנו קשה מאוד, ולקח לנו הרבה זמן עד שהצלחנו להפוך את המימוש למימוש לא רקורסיבי, ואחרי הרבה נסיונות והרבה כתיבת קודים ומחיקתם, הצלחנו להתגבר על האתגר הקשה הזה ולממש את האלגוריתם בצורה רגילה וסוף סוף הוא לוקח זמן הגיוני יחסית לאלגוריתם זה שגם כך הוא לא האלגוריתם הכי טוב מבחינת זמן ריצה.

### -4 BIA\* Algorithm :

אלגוריתם נחמד ויפה – הוא אלגוריתם שדרש הרבה זמן להבנת תנאי העצירה שלו – בהתחלה לא הבנו ממש את תנאי העצירה והיינו עוצרים אחרי שמוצאים צומת משני הטורים, אבל לצערינו שלא תמיד היינו מקבלים תוצאה אופטימלית – ורק אחרי שחזרנו על ההרצאות כמה פעמים וקראנו קצת בספר – הצלחנו להבין שאנחנו לא אמורים לעצור בשלב זה – אלה אחרי שמוצאים צומת מה FORWARDOPEN, BACKWARDOPEN אנחנו נמשיך את החיפוש עד שנקבל צומת שיש לו עלות מינמלית ב BACKWARDOPEN עם FORWARDCLOSED או FORWARDOPEN עם BACKWARDCLOSED.

גם באלגוריתם זה חישנו את היוריסטקה שלנו משני הכיוונים – כלומר אחת FORWARD שמעריכה מההתחלה עד הסוף והשניה BACKWARD שמעריכה מהסוף להתחלה . באלגוריתם זה מימשנו את ה OPENFORWARD ו OPENBACKWARD באמצעות ערימת מינימום כי ראינו שך אנחנו מקבלים זמן ריצה יותר טוב מהשימוש במערך .

#### 5- IDS Algorithm

אלגוריתם חיפוש לא מיועד , והוא שילוב של BFS עם DFS . באלגוריתם זה השתמשנו בשתי פונקציות – הפונקציה הראשית IDS שיש בה לולאה D שרצה מ 1 עד אינסוף , והיא קוראת לפונקציה שניה DLS . בפונקציה ה DLS אנחנו מריצים DFS עד העומק D ואז אנחנו מחזירים אם מצאנו פתרון עד העומק הזה . אחד הדברים המעניינים שהשתמשנו בהם באלגוריתם זה – הוא המילון = DICTIONARY במילון זה אנחנו שומרים את כל הצמתים שבקרנו בהם כבר כדי שלא נחזור ונפרוש אותם שוב פעם --- וזה דבר שהיה מאוד חשוב לעשות ולקח לנו מלא זמן עד שהבנו את החשיבות שלו , כי לפני שנעשה את הדבר הזה האלגוריתם היה לוקח הרבה זמן – כי הוא היה פורש את הצמתים שוב ושוב ( יש צמתים שהיה חוזר ופורש אותם מחדש ) . אחרי שהוספנו את המילון הזה , קיבלנו שיפור טוב מאוד מבחינת זמן ריצה .

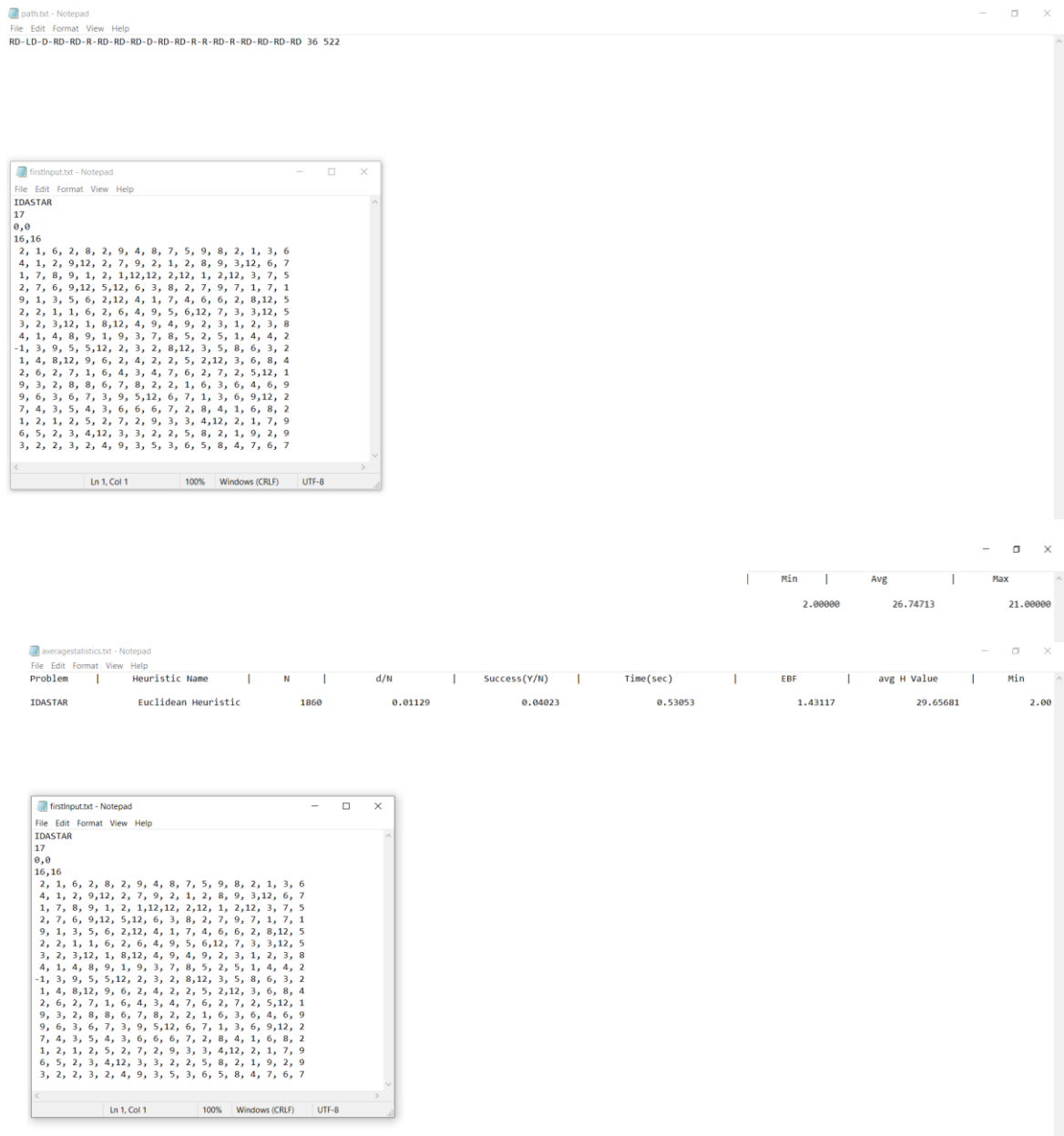
**\*\* דבר חשוב מאוד עבורנו שרוצים לדבר עליו , והוא אחד היתרונות שהרווחנו מהעבודה על הפרויקט שלא קשור ספציפי לנושא עצמו , הוא אופן העבודה המשותף בין שני השותפים ואיך היינו משתפים פעולה .**  
בהתחלה עבדנו ביחד על הפרויקט – עד שתכננו את הבסיס ואיך אנחנו רוצים לעבוד , ואחר כך היינו עובדים כל אחד בנפרד על מימוש אלגוריתם מסויים , עד שמישהו יצליח לסיים את המימוש – אז עד שאחד יתייחס ☺ למשל ב IDA \* , עבדנו מלא זמן ואז בסוף עבדנו ביחד דרך הזום כדי שנצליח להתגבר על הבעיות שהיו לנו . ידענו גם שאנחנו חייבים ללמוד איך להשתמש ב GitHub - לכן בהתחלה למדנו טוב והכרנו איך נעזרים בכלי החשוב הזה – ורק אז התחלנו לעבוד על הפרויקט ולשתף את הקודים שלנו באמצעות ה GitHub

- הערה: לא לקחנו בחשבון את העלות של הצומת הראשונה במסלול.
- מצורף את הקישור של הפרויקט שלנו:

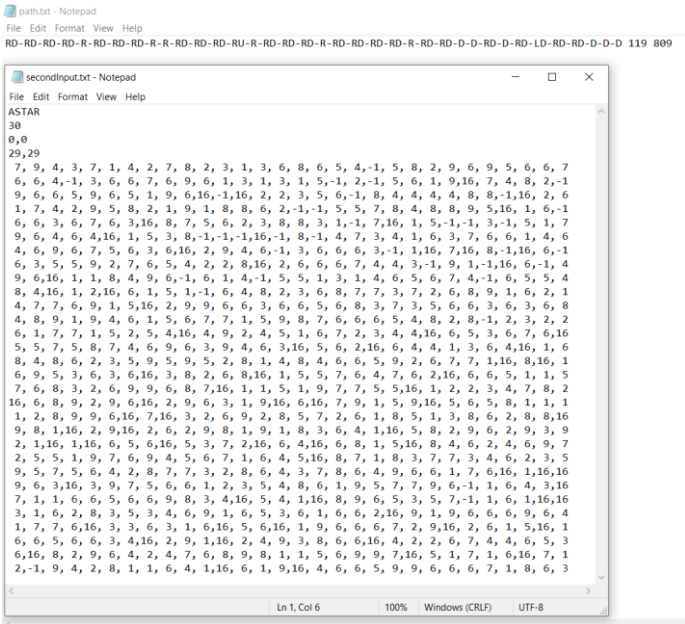
<https://github.com/shadihalloun35/ArtificialIntelligenceProject.git>

## תוצאות הניסוי:

## דוגמא ראשונה:

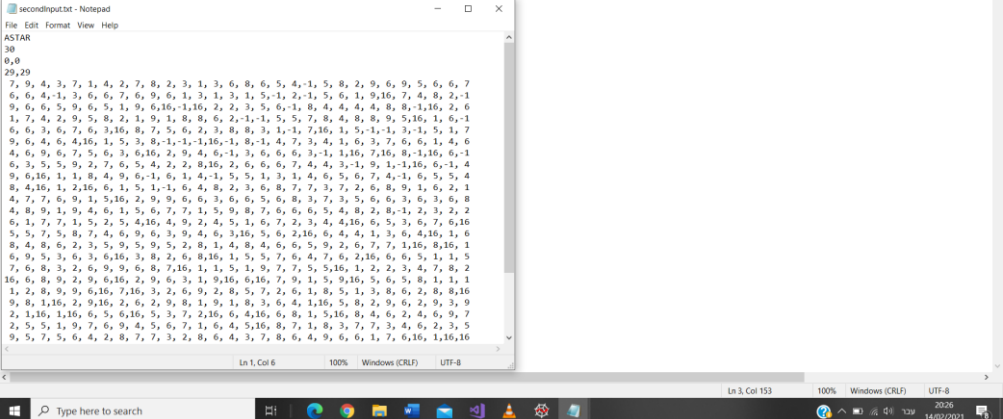


## דוגמא שנייה:



Min	Avg	Max
2.00000	21.70952	40.00000

Problem	Heuristic Name	N	d/N	Success(Y/N)	Time(sec)	EBF	avg H Value	Min
ASTAR	Euclidean Heuristic	1055	0.03791	0.04607	1.27778	1.20104	126.04965	2.000



## דוגמא שלישית:

thirdInput.txt - Notepad

File Edit Format View Help

BIASTAR

60

0,0

59,59

7, 9, 4, 3, 7, 1, 4, 2, 7, 8, 2, 3, 1, 3, 6, 8, 6, 5, 4, -1, 5, 8, 2, 9, 6, 9, 5, 6, 6, 7, 7, 9

6, 6, 4, -1, 3, 6, 6, 7, 6, 9, 6, 1, 3, 1, 3, 1, 5, -1, 2, -1, 5, 6, 1, 9, 16, 7, 4, 8, 2, -1, 6, 6,

9, 6, 6, 5, 9, 6, 5, 1, 9, 6, 16, -1, 16, 2, 2, 3, 5, 6, -1, 8, 4, 4, 4, 4, 8, 8, -1, 16, 2, 6, 9, 6,

1, 7, 4, 2, 9, 5, 8, 2, 1, 9, 1, 8, 8, 6, 2, -1, -1, 5, 5, 7, 8, 4, 8, 8, 9, 5, 16, 1, 6, -1, 9, 6,

6, 6, 3, 6, 7, 6, 3, 16, 8, 7, 5, 6, 2, 3, 8, 8, 3, 1, -1, 7, 16, 1, 5, -1, -1, 3, -1, 5, 1, 7, 9, 6,

9, 6, 4, 6, 4, 16, 1, 5, 3, 8, -1, -1, -1, 16, -1, 8, -1, 4, 7, 3, 4, 1, 6, 3, 7, 6, 6, 1, 4, 6, 9, 6,

4, 6, 9, 6, 7, 5, 6, 3, 6, 16, 2, 9, 4, 6, -1, 3, 6, 6, 6, 3, -1, 1, 16, 7, 16, 8, -1, 16, 6, -1, 9, 6,

6, 3, 5, 5, 9, 2, 7, 6, 5, 4, 2, 2, 8, 16, 2, 6, 6, 6, 7, 4, 4, 3, -1, 9, 1, -1, 16, 6, -1, 4, 9, 6,

9, 6, 16, 1, 1, 8, 4, 9, 6, -1, 6, 1, 4, -1, 5, 5, 1, 3, 1, 4, 6, 5, 6, 7, 4, -1, 6, 5, 5, 4, 9, 6,

8, 4, 16, 1, 2, 16, 6, 1, 5, 1, -1, 6, 4, 8, 2, 3, 6, 8, 7, 7, 3, 7, 2, 6, 8, 9, 1, 6, 2, 1, 9, 6,

4, 7, 7, 6, 9, 1, 5, 16, 2, 9, 9, 6, 6, 3, 6, 6, 5, 6, 8, 3, 7, 3, 5, 6, 6, 3, 6, 3, 6, 8, 9, 6,

4, 8, 9, 1, 9, 4, 6, 1, 5, 6, 7, 7, 1, 5, 9, 8, 7, 6, 6, 6, 5, 4, 8, 2, 8, -1, 2, 3, 2, 2, 9, 6,

6, 1, 7, 7, 1, 5, 2, 5, 4, 16, 4, 9, 2, 4, 5, 1, 6, 7, 2, 3, 4, 4, 16, 6, 5, 3, 6, 7, 6, 16, 9, 6,

5, 5, 7, 5, 8, 7, 4, 6, 9, 6, 3, 9, 4, 6, 3, 16, 5, 6, 2, 16, 6, 4, 4, 1, 3, 6, 4, 16, 1, 6, 9, 6,

8, 4, 8, 6, 2, 3, 5, 9, 5, 9, 5, 2, 8, 1, 4, 8, 4, 6, 6, 5, 9, 2, 6, 7, 7, 1, 16, 8, 16, 1, 9, 6,

6, 9, 5, 3, 6, 3, 6, 16, 3, 8, 2, 6, 8, 16, 1, 5, 5, 7, 6, 4, 7, 6, 2, 16, 6, 6, 5, 1, 1, 5, 9, 6,

7, 6, 8, 3, 2, 6, 9, 9, 6, 8, 7, 16, 1, 1, 5, 1, 9, 7, 7, 5, 5, 16, 1, 2, 2, 3, 4, 7, 8, 2, 9, 6,

16, 6, 8, 9, 2, 9, 6, 16, 2, 9, 6, 3, 1, 9, 16, 6, 16, 7, 9, 1, 5, 9, 16, 5, 6, 5, 8, 1, 1, 1, 9, 6,

1, 2, 8, 9, 9, 6, 16, 7, 16, 3, 2, 6, 9, 2, 8, 5, 7, 2, 6, 1, 8, 5, 1, 3, 8, 6, 2, 8, 8, 16, 9, 6,

9, 8, 1, 16, 2, 9, 16, 2, 6, 2, 9, 8, 1, 9, 1, 8, 3, 6, 4, 1, 16, 5, 8, 2, 9, 6, 2, 9, 3, 9, 2, 1,

2, 1, 16, 1, 16, 6, 5, 6, 16, 5, 3, 7, 2, 16, 6, 4, 16, 6, 8, 1, 5, 16, 8, 4, 6, 2, 4, 6, 9, 7, 2, 5,

2, 5, 5, 1, 9, 7, 6, 9, 4, 5, 6, 7, 1, 6, 4, 5, 16, 8, 7, 1, 8, 3, 7, 7, 3, 4, 6, 2, 3, 5, 2, 5,

9, 5, 7, 5, 6, 4, 2, 8, 7, 7, 3, 2, 8, 6, 4, 3, 7, 8, 6, 4, 9, 6, 6, 1, 7, 6, 16, 1, 16, 16, 2, 5

Ln 1, Col 1 100% Windows (CRLF) UTF-8

averagestatistics.txt - Notepad

File Edit Format View Help

Problem	Heuristic Name	N	d/N	Success(Y/N)	Time(sec)	EBF	avg H Value	Min
BIASTAR	Euclidean Heuristic	2573	0.01982	0.02989	2.86277	1.16646	310.89685	2.0

thirdInput.txt - Notepad

File Edit Format View Help

BIASTAR

60

0,0

59,59

7, 9, 4, 3, 7, 1, 4, 2, 7, 8, 2, 3, 1, 3, 6, 8, 6, 5, 4, -1, 5, 8, 2, 9, 6, 9, 5, 6, 6, 7, 7, 9

6, 6, 4, -1, 3, 6, 6, 7, 6, 9, 6, 1, 3, 1, 3, 1, 5, -1, 2, -1, 5, 6, 1, 9, 16, 7, 4, 8, 2, -1, 6, 6,

9, 6, 6, 5, 9, 6, 5, 1, 9, 6, 16, -1, 16, 2, 2, 3, 5, 6, -1, 8, 4, 4, 4, 4, 8, 8, -1, 16, 2, 6, 9, 6,

1, 7, 4, 2, 9, 5, 8, 2, 1, 9, 1, 8, 8, 6, 2, -1, -1, 5, 5, 7, 8, 4, 8, 8, 9, 5, 16, 1, 6, -1, 9, 6,

6, 6, 3, 6, 7, 6, 3, 16, 8, 7, 5, 6, 2, 3, 8, 8, 3, 1, -1, 7, 16, 1, 5, -1, -1, 3, -1, 5, 1, 7, 9, 6,

9, 6, 4, 6, 4, 16, 1, 5, 3, 8, -1, -1, -1, 16, -1, 8, -1, 4, 7, 3, 4, 1, 6, 3, 7, 6, 6, 1, 4, 6, 9, 6,

4, 6, 9, 6, 7, 5, 6, 3, 6, 16, 2, 9, 4, 6, -1, 3, 6, 6, 6, 3, -1, 1, 16, 7, 16, 8, -1, 16, 6, -1, 9, 6,

6, 3, 5, 5, 9, 2, 7, 6, 5, 4, 2, 2, 8, 16, 2, 6, 6, 6, 7, 4, 4, 3, -1, 9, 1, -1, 16, 6, -1, 4, 9, 6,

9, 6, 16, 1, 1, 8, 4, 9, 6, -1, 6, 1, 4, -1, 5, 5, 1, 3, 1, 4, 6, 5, 6, 7, 4, -1, 6, 5, 5, 4, 9, 6,

8, 4, 16, 1, 2, 16, 6, 1, 5, 1, -1, 6, 4, 8, 2, 3, 6, 8, 7, 7, 3, 7, 2, 6, 8, 9, 1, 6, 2, 1, 9, 6,

4, 7, 7, 6, 9, 1, 5, 16, 2, 9, 9, 6, 6, 3, 6, 6, 5, 6, 8, 3, 7, 3, 5, 6, 6, 3, 6, 3, 6, 8, 9, 6,

4, 8, 9, 1, 9, 4, 6, 1, 5, 6, 7, 7, 1, 5, 9, 8, 7, 6, 6, 6, 5, 4, 8, 2, 8, -1, 2, 3, 2, 2, 9, 6,

6, 1, 7, 7, 1, 5, 2, 5, 4, 16, 4, 9, 2, 4, 5, 1, 6, 7, 2, 3, 4, 4, 16, 6, 5, 3, 6, 7, 6, 16, 9, 6,

5, 5, 7, 5, 8, 7, 4, 6, 9, 6, 3, 9, 4, 6, 3, 16, 5, 6, 2, 16, 6, 4, 4, 1, 3, 6, 4, 16, 1, 6, 9, 6,

8, 4, 8, 6, 2, 3, 5, 9, 5, 9, 5, 2, 8, 1, 4, 8, 4, 6, 6, 5, 9, 2, 6, 7, 7, 1, 16, 8, 16, 1, 9, 6,

6, 9, 5, 3, 6, 3, 6, 16, 3, 8, 2, 6, 8, 16, 1, 5, 5, 7, 6, 4, 7, 6, 2, 16, 6, 6, 5, 1, 1, 5, 9, 6,

7, 6, 8, 3, 2, 6, 9, 9, 6, 8, 7, 16, 1, 1, 5, 1, 9, 7, 7, 5, 5, 16, 1, 2, 2, 3, 4, 7, 8, 2, 9, 6,

16, 6, 8, 9, 2, 9, 6, 16, 2, 9, 6, 3, 1, 9, 16, 6, 16, 7, 9, 1, 5, 9, 16, 5, 6, 5, 8, 1, 1, 1, 9, 6,

1, 2, 8, 9, 9, 6, 16, 7, 16, 3, 2, 6, 9, 2, 8, 5, 7, 2, 6, 1, 8, 5, 1, 3, 8, 6, 2, 8, 8, 16, 9, 6,

9, 8, 1, 16, 2, 9, 16, 2, 6, 2, 9, 8, 1, 9, 1, 8, 3, 6, 4, 1, 16, 5, 8, 2, 9, 6, 2, 9, 3, 9, 2, 1,

2, 1, 16, 1, 16, 6, 5, 6, 16, 5, 3, 7, 2, 16, 6, 4, 16, 6, 8, 1, 5, 16, 8, 4, 6, 2, 4, 6, 9, 7, 2, 5,

2, 5, 5, 1, 9, 7, 6, 9, 4, 5, 6, 7, 1, 6, 4, 5, 16, 8, 7, 1, 8, 3, 7, 7, 3, 4, 6, 2, 3, 5, 2, 5,

9, 5, 7, 5, 6, 4, 2, 8, 7, 7, 3, 2, 8, 6, 4, 3, 7, 8, 6, 4, 9, 6, 6, 1, 7, 6, 16, 1, 16, 16, 2, 5

Ln 1, Col 1 100% Windows (CRLF) UTF-8