



שם התכנית : **תכנון מערכות שפת C** -

מקצוע : **הנדסת תוכנה**

כיתה : **ר"ג**



תוכנית לימודים : תכנון מערכות בשפת C

היקף התוכנית: 196 שעות

הזמן הנדרש:

• עיוני 140 שעות

• מעשי 56 שעות

דרישות קדם

הידע המוקדם ללימוד הפרק הוא הכרת מדעי המחשב בהיקף של 5 יח"ל מלימודי התיכון. תלמידים שאין להם הידע המוקדם יצטרכו להשלימו; חומר זה לא יילמד במסגרת קורס זה.

אוכלוסיית יעד

כיתות י"ג במגמת הנדסת תוכנה.

סביבת העבודה

שפת התכנות האפשרית בקורס זה היא C.

מטרות הקורס

- להקנות מושגי יסוד ועקרונות שעליהם מושתת תחום מדעי המחשב.
- ללמד עקרונות של החשיבה הפרוצדורלית.
- פתרון בעיות באמצעות אלגוריתמים.
- ליישם את המושג אלגוריתם על ידי כתיבת תכניות בשפת תכנות עילית C • והרצתן על מחשב.
- להעמיק ולהרחיב את החומר שנלמד במקצוע מדעי המחשב • בתיכון ולהעמיק את הכתיבה
- והמיומנות בשפת C.
- להוסיף כלים לפיתוח ולמימוש אלגוריתמים, למשל פרוצדורות/פעולות, רקורסיה, מערכים.
- רב ממדיים, מבנים, מערכים דינמיים, הקצאת זיכרון דינמית וטיפול בקבצים.
- להעמיק את הדיון בנושאי נכונות האלגוריתם.
- להציג אלגוריתמים לבעיות חשובות (למשל מיון), המשמשות אמצעי למידה והדגמה.
- לנושאים שנלמדו.
- להעמיק את הדיון בנושא יחידת ספריה וחלוקת התוכנית למודולים.



טבלת הפרקים וחלוקת השעות המוצעת

פרקי הלימוד	עיוני	מעשי
פרק 1 - מודל חישוב בסיסי	10	2
פרק 2 – קלט/פלט בשפת C	4	2
פרק 3 – מבוא לפיתוח אלגוריתמים	4	-
פרק 4 – ביצוע מותנה	4	2
פרק 5 – נכונות האלגוריתם	4	-
פרק 6 – ביצוע חוזר לולאות ומבני בקרה	6	2
פרק 7 – תת משימות, פונקציות/פעולות	7	4
פרק 8 – מערכים חד מימדיים	4	2
פרק 9 – מערכים דו מימדיים ורב מימדיים	4	2
פרק 10 – מצביעים ומימוד פרוצדורות ב- C	14	4
פרק 11- תווים ומחרוזות	8	4
פרק 12 – בעיות אלגוריתמיות מתקדמות	4	2
פרק 13 – טיפוס נתונים מבנה ואיגוד	6	4
פרק 14 – יחידת ספרייה ומאקרו	4	2
פרק 15 – הקצאת זכרון דינמית ומערך דינמי	10	4
פרק 16 – סוגי קבצים, קלט/פלט בקבצים	6	4
פרק 17 – טיפול בסיביות (bits)	4	4
פרק 18 – מבוא לרקורסיה	7	4
פרק 19 – שילוב טיפוסים נתונים ופרוייקטים	20	8
מבחנים	10	-
סה"כ	140	56



פרק 1 : מודל חישוב בסיס (12 שעות)

מטרת הפרק

- להכיר את מודל החישוב של משתנים ושינוי ערכים ואת האלגוריתם הסדרתי הפשוט במודל זה.
- הכרת אבני היסוד של שפת C.

פירוט התכנים

נתונים; משתנים; אלגוריתם פשוט כסדרה של הוראות לשינוי ערכים; תהליך הביצוע; מצבים בזמן ביצוע; פיתוח אלגוריתם "שטוח" (זיהוי קלטים ופלטים, זיהוי ותיעוד משתנים, בחירת טיפוסים משתנים).

טיפוסי נתונים: שלמים, ממשיים, תווים, enum, גדלי הטיפוסים והגדרת טיפוס החדש ע"י typedef.

קבועים, ליטרלים, אופרטורים חשבוניים, אופרטורים יוניים, אופרטורים וביטויים לוגיים ואופרטורים הפועלים על סיביות. אופרטור sizeof אופרטורי הצבה וביטויי הצבה מקוצרים.

המרת טיפוסים מרומזת ומפורשת casting

פונקציות/פעולות ספריה בסיסיות: פונקציות מתימטיות, atoi, atof, log, log10, abs, sqrt, pow, exp - ceil, floor

מספר אקראי. srand, rand – סעיף זה אינו עומד בפני עצמו אלא כמשולב תוכן.

היכרות עם סביבת העבודה של שפת התכנות C -עבודה בסביבת העבודה: טיפוסים משתנים, הגדרת משתנים, הוראות השמה, ביטויים חשבוניים, הוראות קלטים ופלט, מבנה תכנית בסיסית. בדוגמאות יהיו גם אלגוריתמים מילוליים וגם קטעי תכנית. ביצוע תכנית יודגם באמצעות טבלאות מעקב.



פרק 2: קלט/פלט בשפת C (6 שעות)

מטרות הפרק

- לחשוף את התלמידים לקלט/פלט של תו בודד ולפי תבניות.
- להדגים את ניתוב קלט/פלט ☐.

פירוט התכנים

קלט/פלט של תו בודד, קריאת תווי קלט וניתוחם, פלט לפי תבנית ע"י, printf קביעת תצורת הפלט, קלט לפי תבנית ע"י, scanf מציניי הטיפוס של הטיפוסים: תו, שלם עשרוני, שלם בלתי מסומן, שלם בבסיס אוקטלי, שלם בבסיס הקסדצימלי, ממשי בכתיב עשרוני ומעריכי, מחרוזת, ניתוב קלט/פלט.



פרק 3: מבוא לפיתוח אלגוריתמים (4 שעות)

מטרות הפרק

- להמחיש את הצורך במבני בקרה ובאלגוריתמים מורכבים, ותוך כדי כך לתרגל פירוק בעיה
- לתת בעיות פשוטות יותר .

פירוט התכנים

פירוק בעיה לתת בעיות, מנגנוני בנייה של אלגוריתמים מורכבים מאלגוריתמים פשוטים יותר- .
הערה: תוצגנה דוגמאות המראות שלא תמיד די באלגוריתם פשוט (סדרתי). יודגם הקשר הקיים
בין התת בעיות שבפירוק לבין חלקי האלגוריתם.

פרק 4: ביצוע מותנה (6 שעות)

מטרות הפרק

- ללמוד את מבנה הבקרה של ביצוע מותנה בצורות שונות ומרכיביו .
- להכיר אלגוריתמים שמשמשים בהם בביצוע מותנה .

פירוט התכנים

תנאי; ביצוע מותנה; קשר, and קשר or וקשר ; not טבלאות אמת ל or , and -ו ; not - תנאי בוליאני עם יחסי
סדר; משפט (if בגרסה מפוצלת ובגרסה עוקפת); תרגול בהבנת תכניות וכתיבתן.
הדגש יושם על המושגים: התניה , פיצול ועקיפה, ועל העובדה שלא כל קטעי האלגוריתם מבוצעים בכל הרצה .
בדוגמאות ובתרגילים יופיעו תנאים המכילים and , or -ו not
משפט. switch-case



פרק 5: נכונות של אלגוריתמים (4 שעות)

מטרות הפרק

- להכיר בהכרה ראשונית את המושג נכונות שילווה את חומר הלימוד כולו .
- להכיר את היתרונות והחסרונות של בדיקה באמצעות הרצה של אלגוריתם/תכנית .
- להכיר את החשיבות של בניית אלגוריתם בצורה נאותה ושל תיעוד הולם (כולל תיעוד הקלטים החוקיים).
- להכיר את ההבדל בין שגיאה לוגית באלגוריתם לבין שגיאה הנובעת מטעות במימוש האלגוריתם בשפת התכנות.
- תלמידי הרמה הרגילה יתמקדו בבדיקת הפלט עבור דוגמאות שונות של קלט, הפקת פלט ע"פ קלט נתון , אימות האלגוריתם ע"י טבלת מעקב, אפיון קלטים לפלט נתון.

פירוט התכנים

תחום הקלטים החוקיים; נכונות ביחס לבעיה אלגוריתמית וקלטיה; אלגוריתם נכון ואלגוריתם שגוי; בבדיקת אלגוריתם באמצעות הרצה על קלטי בדיקה ומגבלותיה של בדיקה כזאת; תיעוד ככלי עזר להשגת נכונות ובדיקת נכונות.

תכנית שגויה (שגיאות תחביר, שגיאות בזמן ריצה, פלטים שגויים).
הרצה ובדיקה של תכניות קיימות.



פרק 6: ביצוע חוזר לולאות ומבני בקרה (8 שעות)

מטרות הפרק

- ללמוד את מבנה הבקרה של ביצוע חוזר.
- לתרגל ביצוע חוזר ככלי לעידון אלגוריתמים.
- להבחין בין כתיבה אלגוריתמית של לולאה לבין מימושה בשפת תכנות.
- להשתמש בלולאות לאלגוריתמים המצריכים מנייה או צבירה.

פירוט התכנים

ביצוע חוזר: מקרים נפוצים (פעולה אחת החוזרת על עצמה, פעולה החוזרת ומתבצעת על נתונים שונים, ביצוע חוזר עם פעילות מצטברת); מונים; צוברים; תנאי סיום; ביצוע אינסופי. משפט, while, תנאי בוליאני כזקיף. הערות: הצד היישומי בפרק זה יתרכז במשפט ה-, do-while, while, ו- for .. - יש להדגים גם באמצעות פסאודו קוד נייטרלי שאין לו תרגום ישיר לשפת התכנות, למשל: "בצע לכל איבר- בקבוצה". במקרים כאלה ניתן לתרגל גם באמצעים לא ממוחשבים. יש להדגיש שבביצוע חוזר טקסט האלגוריתם נשאר קבוע, אך הוא משרה תהליכי ביצוע באורכים שונים כולל אורך התלויים בקלט. יש לקשור זאת לביצוע שאינו מסתיים (דהיינו בעל אורך אינסופי). שימוש נכון בזקיף, לולאות מקוננות ומשפטי switch-case אין להשתמש בהוראות break, continue.

פרק 7 : תת משימות: פונקציות (11 שעות)

מטרות הפרק

- ללמוד את מושג הפונקציה/הפעולה כאמצעי לפתרון בעיה בעזרת פתרון תת בעיות.
- ליישם פונקציות/פעולות בשפת התכנות.
- להכיר סוגים שונים של משתנים ושל פרמטרים.

פירוט התכנים

שימוש בפונקציה/בפעולה כהתמודדות עם תת בעיה; אלגוריתם לפונקציה/לפעולה כפתרון התת-בעיה; קריאה לפונקציה/לפעולה כהוראה באלגוריתם הקורא.



פונקציות/פעולות בשפת התכנות: הגדרת פונקציה/פעולה, גוף הפונקציה/הפעולה, פרמטר ערך, משתנה מקומי, מקומי סטטי וגלובלי. משתנה מקומי בבלוק וערך מוחזר.
תחום פעולה (scope) ומשך ה"חיים" של משתנה.
פונקציות שאינן מחזירות ערך.
מנגנון הפונקציות ב-C. הכרזה על האבטיפוס פונקציה.
תיעוד של פונקציה / פעולה, כולל ציון טענות הכניסה ויציאה לפרוצדורה / פונקציה / פעולה.
מבנה זיכרון התכנית: קטע הקוד, קטע הנתונים ומחסנית הקריאות.
הערות: יש לקשור את החומר עם המושגים העיוניים של פרק 3. יש לשלב פונקציות/פעולות הנכתבות על ידי התלמידים עם פונקציות/פעולות מערכת. יש להדגיש את הצורך בנכונות- הפונקציה/הפעולה הנקראת ביחס להנחות הקלט שלה.

פרק 8: מערכים חד ממדיים (6 שעות)

מטרות הפרק

- להכיר את הצורך במבני נתונים.
- ללמוד את מבנה הנתונים מערך כאוסף לינארי של משתנים מאותו הסוג.

פירוט התכנים

מבני נתונים והצורך בהם; מערך; מציין: הגדרת מערך; טיפוס המערך; מציינים; טווח המציינים; טיפוס מציינים.
טיפוס תווי ומערך תווי.
צורות שונות של אתחול מערך ב C-והעברת מערך כפרמטר לפונקציה ושיקולי מודולריות.
הערות: יש להדגיש את ההקבלה שבין משתנה לתא של מערך, ואת העובדה שמערך נבנה כדי לאסוף יחד קבוצה של משתנים למטרה משותפת ולאפשר טיפול אלגוריתמי אחיד. יש להראות את הקשר ההדוק שבין ביצוע חוזר כמנגנון בנייה של אלגוריתם לבין מערך כמנגנון ארגון של נתונים. יש להדגיש את הקשר שבין משפט for למערך חד ממדי ואת תפקידו הכפול של המציין- בהקשר זה.



פרק 9: מערכים דו ממדיים ורב ממדיים (6 שעות)

מטרות הפרק

- להכיר את הצורך במבני נתונים מערך דו/רב ממדי .
- ללמוד את מבנה הנתונים מערך כאוסף ליניארי של משתנים מאותו הסוג.

פירוט תכנים

הגדרת מערך דו ממדי; מציינים למערך דו ממדי. פתרון בעיות מורכבות באמצעות שילוב והרכבה של מבני בקרה ומערכים דו/רב ממדיים. קיר ביטחון והיתרונות בשימוש קיר ביטחון. הערה: יש להדגים ולתרגל את הנלמד באמצעות בעיות מתקדמות ומורכבות יותר (לדוגמה: בעיות על מערכים ממוינים או כמעט ממוינים) (לא מיון), חיפוש סדרתי, מציאת זוגות שאינם בסדר הנכון וכדומה.)חיפוש בינארי. ניתן להרחיב בנושא של תווים בעזרת דוגמאות מעולם עיבוד התמלילים. יש להעמיק ולהרחיב גם בנושאי יעילות ובסוגים שונים של ביצועים חוזרים ושילובם.

פרק 10: מצביעים(מחווים) והחזרת ערכים על ידי פונקציות שאינן מחזירות ערך (18 שעות)

מטרות הפרק

- להבין מהו מצביע ושהפונקציה יחד עם המצביע היא אמצעי נוסף לפתרון בעיה בעזרת תתי בעיות .
- להעמיק את הידע בנושא פונקציות שאינן מחזירות ערך ופונקציות שכן מחזירות ערך וההבדלים ביניהן.

פירוט התכנים

הגדרת מצביע, גישה למשתנה דרך המצביע אליו, מצביע למצביע, העברת פרמטר לפונקציה ע"י מצביע. שימוש בפונקציות שאינן מחזירות ערך כפתרון לתת בעיה; סוגי פרמטרים; אבחנה בין call by value - לבין call by reference . הפונקציה / פעולה; קריאה לפונקציה / פעולה; חזרה על פרמטר ערך; פרמטר משתנה; scope של משתנה; תיעוד של פונקציה/ פעולה, כולל ציון טענות הכניסה ויציאה לפרוצדורה / פונקציה / פעולה. קשר בין מצביעים למערכים: שקילות בין מערך למצביע, הבדלים בין מצביע למערך, פעולות חשבוניות על מצביעים, מצביעים לפונקציות.



שימושים במצביעים לפונקציות תקניות - C: qsort , חיפוש בינארי. bsearch
מערכי מצביעים ומערכי מצביעים לפונקציות.
מחוננים לפונקציות.

פרק 11: תווים ומחרוזות (12 שעות)

מטרת הפרק

- להקנות כלים לפתרון בעיות לעיבוד טקסט.

פירוט התכנים

מחרוזות; יחס סדר מילוני; ייצוג מחרוזות כמערכי תווים; בניית אלגוריתמים בעבור מחרוזות.
המחרוזות היא מערך של תווים; אורך מחרוזת, השוואת מחרוזות, העתקת מחרוזות, מחיקת מחרוזות, שרשור
מחרוזות, מציאת תת מחרוזת- .
קלט/פלט של מחרוזות, פונקציות לטיפול במחרוזות strcmp , strcpy , sscanf , sprintf , strlen , strcmp
strcat ,
פונקציות לפעולות חיפוש טקסט strstr , strchr , strchr , strspn , strtoc ,
פונקציות לבדיקת סוג התו. isalnum, isalpha, islower, isupper, isdigit, isxdigit :
מערכי מחרוזות, מערך מצביעים למחרוזות לעומת מערך של מערכים, מיון מערך מחרוזות.

פרק 12: בעיות אלגוריתמיות מתקדמות (6 שעות)

מטרות הפרק

- להכיר את בעיית המיון והמיזוג ואלגוריתמים שונים לפתרון .
- לתרגל ולהעמיק בנושאים הנלמדים ביחידה.

פירוט התכנים

חיפוש ומיון, מימוש פעולת חיפוש: סידרתי ובינארי, מימוש פעולת מיון; מיזוג מערכים ממוינים; בעיות נוספות ,
כולל כאלה המשתמשות במספרים אקראיים.
הערות: יש לשלב כאן פרוצדורות/פונקציות/פעולות מערכת נוספות שטרם נלמדו ככלי עזר לפתרון הבעיות הנ"ל.
יש ללמד לפחות שתי שיטות מיון, מיון הכנסה ומיון בועות.



פרק 13: טיפוסים נתונים מבנה ואיגוד (10 שעות)

מטרות הפרק

- להכיר את המושג "טיפוס נתונים"
- להכיר את מבנה הנתונים "רשומה", "מבנה"
- לבנות מבנים מורכבים.

פירוט התכנים

המושג טיפוס: הצהרת טיפוס, טיפוס מפורט.

מבני נתונים מורכבים: מבנים רשומות ; שדה ברשומה; הגדרת רשומות; גישה לשדה בתוך רשומה; קליטה לתוך רשומה. מבנים בתוך מבנים, מערכי מבנים, מצביעים למבנים, העברת מבנים כפרמטר לפונקציות, מיון מערך מבנים, מטריצות של מבנים.
איגוד . Union - מערך של איגודים.

פרק 14: יחידת ספרייה ומאקרו (6 שעות)

מטרות הפרק

- לערוך היכרות מעמיקה עם יחידת הספרייה בסביבת העבודה - C
- לחדד את רעיון המודולריות של מערכת תכנה באמצעות בנייה של יחידות ספרייה ושימוש בהן.

פירוט התכנים

חלוקת התוכנה למספר קבצים, קבצי ממשק וקבצי מימוש, הכללת קבצי ממשק, בניית יחידת ספרייה בסביבת העבודה; משתנה פנימי; שגרה פנימית; קישור יחידת ספרייה לתכנית.
הקדם מעבד : (Pre-Processor) - הגדרת קבועים, הגדרת פונקציות מאקרו, הידור מותנה.



פרק 15 : הקצאת זיכרון דינמית ומערך דינמי (14 שעות)

מטרות הפרק

- לערוך היכרות מעמיקה עם הקצאת זיכרון בזמן ריצה .
- לחדד את ההבדלים בין זמן ריצה לבין זמן הידור (קומפילציה) .

פירוט התכנים

הקצאת מקום למשתנים ושיקולי צריכת זיכרון, מנגנון הקצאת זיכרון דינמית, הקצאה ע"י malloc ו – calloc ושחרור ע"י free, הקצאה ע"י strdup מערכי מצביעים, מיון מערך מצביעים, מערך דינמי והרחבת המערך ע"י realloc

פרק 16: קלט/פלט בקבצים (10 שעות)

מטרות הפרק

- לערוך היכרות מעמיקה עם עבודה בקבצים .
- לחדד את ההבדלים בין קובץ מסוג טקסט לבין קובץ בינארי

פירוט התכנים

סוגי קבצים, פתיחת קובץ, קריאת וכתובת טקסט לקובץ, קריאה וכתובת לקובץ בינארי, חיפוש בקובץ בינארי, מימוש האלגוריתם לחיפוש בינארי בקובץ בינארי, מימוש האלגוריתם לחיפוש סידרתי בקובץ בינארי, מיון ומיזוג קבצים.



פרק 17: טיפול בסיביות (bits) - (8 שעות)

מטרות הפרק

- לערוך היכרות מעמיקה עם עבודה בביטים .

פירוט התכנים

מספרים בעלי סימן וחסרי סימן, פעולות לוגיות בסיביות, -, xor , or , and - משלים ל 1 משלים ל 2 , הזזת סיביות שמאלה וימינה, הזזה אריתמטית, ההבדל בין פעולות לוגיות לבין - פעולות לוגיות בסיביות.

פרק 18 : מבוא לרקורסיה (11 שעות)

מטרות הפרק

- להכיר הכרה ראשונית את המושג רקורסיה ככלי לפתרון בעיות .
- להכיר את היתרונות והחסרונות של כתיבה רקורסיבית
- להכיר את ההבדל בין הגדרה רקורסיבית לבין מימושה .
- להכיר. Backtracking

פירוט התכנים

קריאה רקורסיבית; תנאי עצירה; מעקב על אלגוריתמים רקורסיביים; כתיבת אלגוריתם רקורסיבי. כתיבת אלגוריתם רקורסיבי בשיטת. Backtracking

פרק 19 : ווקטור כיוונים, שילוב טיפוסים נתונים ופרויקטים (28 שעות)

מטרות הפרק

- מהו מערך כיוונים, איך בונים אותו וכיצד משתמשים בו.
- יצירת מבני נתונים מורכבים וביצוע פרויקטים .



פירוט התכנים

פרויקטים המשלבים את כל מבני הנתונים שנלמדו בקורס זה ולהלן מספר דוגמאות:

דוגמא ראשונה- דירה להשכיר

בעמק יפה בין הרים ושדות עומד מגדל בן עשר קומות (שבע דירות בכל קומה), ומי גר במגדל? אז זהו שבינתיים אף אחד. המגדל החדש נבנה לא מזמן ונקנה ע"י חברת ההשכרה טויט ובניו בע"מ. החברה רוצה להשכיר את הדירות בבניין אבל יש לה מספר בעיות לא קטן. מידי פעם מגיע עובר אורח לשכור דירה, הבעיה שכל אחד שמגיע רוצה לגור ליד שכנים שמוצאים חן בעיניו גם מעליו גם מתחתיו וגם מצדדיו. לבסוף הגיע מנהל החברה, בונה מפורסם, למסקנה שאי אפשר להמשיך ככה וחייבים לפתור את בעיית השכנים. ופה אתם נכנסים לפעולה, אתם תכתבו מערכת שתמחשב את תהליך ההשכרה. אז ככה, עבור כל דייר חדש שמגיע יש לקלוט את סוג החיה שלו וליד כמה שכנים הוא מוכן לגור. ישנם 5 סוגי חיות אפשריות ולכל חיה יש קוד משלה.

1 - תרנגולת עצלה

2 - נמלה

3 - חתולה כושית .

4 - חזיר לבנבן .

5 - יונה .

מובטח שהקלט תקין.

עכשיו מגיע החלק המסובך, יש לסרוק את הדירות ולמקם כל דייר שנקלט במקום כלשהו בבניין. אבל כמו בכל דבר בחיים, יש הסתייגויות, ואלו הן:

נמלה לא מסכימה להיות שכנה של תרנגולת עצלה (כל היום על משכבה מתהפכת, היא כל כך שמנה שקשה לה ללכת), כמוכן שהתרנגולת העצלה לא רוצה לגור ליד הנמלה.

חזיר וחתולה כושית לא מוכנים לגור האחד ליד השני (לא נאה לו ולא יאה לו ...) ורק היונה אוהבת את כל שכניה. מה אתם צריכים לעשות?

אתם צריכים למצוא לכל דייר שנקלט דירה שבה אף אחד משמונת שכניו אינו שכן לא רצוי מבחינתו, כמו כן יש לשים לב שמספר השכנים של הדייר לא יעלה על המספר שהוא ביקש. עבור כל דייר יש להדפיס הודעה שמודיעה איפה הדירה החדשה שהוא השכיר

(לדוגמא : דירה 7 בקומה 6), ואם לא נמצאה דירה יש להדפיס הודעה כי חברת טויט ובניו בע"מ מצטערת אבל הוא יצטרך ללכת לחפש דירה במקום אחר.

בסוף התכנית יש להדפיס את מבנה הבניין (מי גר איפה, פשוט ריבוע שמהווה תמונה של הבניין).



קבצים:

קובץ קלט

כמובן שאנחנו לא קולטים ישירות מהמשתמש, חברת ההשכרה שלנו יצרה קובץ מצוין המכיל את כל הדיירים שנכנסים לבניין, הקובץ יהיה בנוי משורה עבור כל דייר במבנה הבא:

AnimalType NumOfNeighbours

השורה הראשונה בקובץ תכיל את שם קובץ הפלט עבור הבניין.

שימו לב שאין אינדיקציה לסוף הקלט למעט. eof

שם קובץ הקלט יהיה. Tenants.dat

קובץ פלט

גם את כל ההדפסות שלנו נכתוב לקובץ פלט.

שם הקובץ ניתן בקובץ הקלט.

ההודעות היחידות שיודפסו למסך הן הודעות על כך שהקבצים לא נפתחו.

דגשים:

ברגע שדייר התמקם אין צורך לשמור על מספר השכנים שהוא ביקש) הוא כבר חתם על החוזה, אם יש לו תלונות לא נורא).

מובטח שמבנה קובץ הקלט תקין, כמו גם הקלט עצמו (כלומר שם קובץ הפלט תקין, מספר השכנים הוא בין 0 ל-8, וסוג החיה תקין).

תחשבו טוב איך לייצג את הבניין וכל דירה בו.

יש לשים לב שהקומות ממוספרות מ 1 עד 10 והדירות מ 1 עד 7 ולא מ0

ומאז ועד היום גרים בבניין שכנים טובים בחיי שלום.

את התרגול יש לכתוב ב שפת C.

יש להשתמש בכל מה שלמדתם בשיעור על קיר ביטחון ווקטור כיוונים.



דוגמא שניה - בולי במבוך החיים

בעקבות הצלחתה הרבה של תכנית הטלוויזיה "החילוץ והצב" החליטו גדי וימימה לרכוש לעצמם את הגרסה הביתית של המשחק "בולי במבוך החיים" שזכתה להצלחה אדירה. אך מה רבה היתה אכזבתם כשהגיעו לחנות "חיים צעצועים וכלי כתיבה בע"מ" והתברר להם שכל המשחקים אזלו...

כשהיה הצמד כבר כמעט על סף ייאוש התפשט לפתע חיוך על פניה של ימימה:

כן... אם זה עבד פעם אין סיבה שזה לא יעבוד שוב!

אכן סטודנטים יקרים, לא טעיתם בכוונותיה של ימימה. אתם תכתבו את המשחק עבור הצמד!

תאור המשימה:

המשחק "בולי במבוך החיים" דומה דמיון מפליא למשחק ("pacman") עד היום יש ויכוחים על זכויות היוצרים.)

המשחק מורכב מלוח משחק בגודל 100 X 100 ובו מבוך. בתוך המבוך ישנן נקודות, אותן צריך לאסוף בולי תוך כדי מעבר במבוך. המשחק מסתיים ברגע שלא נשארו יותר נקודות על הלוח. נשמע פשוט לא? זה באמת פשוט ולכן נסתבך מעט...

ובכן בולי לא נמצא לבדו במבוך יחד איתו נמצא גם דב שבלולים (בטח שיש דבר כזה) שמטרתו (מן הסתם) היא לאכול את בולי לפני שיספיק לאכול את כל הנקודות שבמבוך.

הקלט:

הקלט הראשון הוא מיקום קירות המבוך. הקלט מופיע בצורה של זוגות אינדקסים (שורה וטור) עד שנקלט 0 עבור אחד מהם.

כל זוג מורכב מ:

מספר שורה.

מספר טור.

כל נקודה כזו מציינת חלק מקיר המבוך. שימו לב מובטח שהמשתמש מכניס את ההנחיות בתחום 1-100. לאחר מכן יש לקלוט את המקום ההתחלתי של בולי ואת מיקומו ההתחלתי של הדב (בצורת מס' שורה ומס' טור). יתכן והדב ימוקם באותו מקום כמו בולי. מובטח שהקלט תקין.

העיבוד:

כל נקודה על הלוח שאינה שייכת לקירות המבוך מהווה נקודה שבולי צריך לאסוף. בולי אוסף את הנקודה ע"י מעבר על פניה, לאחר שבולי אוסף את הנקודה היא נמחקת מהלוח.

מהלך המשחק:

צורת התזוזה של בולי:

יש לקלוט קודי תזוזה של בולי ולזוז בהתאם על הלוח. משמעות הקוד:

1 - תזוזה ימינה



2 - תזוזה שמאלה

3- תזוזה למעלה

4 – תזוזה למטה

0 - המשך התזוזה בכיוון הקודם

אופן התנועה:

מצב ראשוני : בולי עומד על משבצת ההתחלה (המיקום ההתחלתי שקלטנו מהמשתמש).
המשחק יתחיל אך ורק עם קליטת קוד שונה מ- 0 (כלומר 1-4). כל עוד לא נקלט קוד כזה, יש להמשיך לקלוט קודים.

עם התחלת התזוזה יש לנוע בהתאם לקוד הנקלט, אחרי כל תנועה יש לקלוט קוד תנועה נוסף. אם נקלט קוד תנועה 0 יש להמשיך את התנועה בכיוון הקודם.

אם השחקן מנסה לנוע אל תוך קיר המבוך או מחוץ ללוח הוא נשאר במקום שהיה לפני קליטת קוד התנועה. צורת התזוזה של דוב השבלולים:

דוב השבלולים שואף להתקדם כמה שיותר קרוב לבולי בקו אווירי (הוא לא מספיק חכם כדי לדעת שקירות המבוך יחסמו אותו). דוב השבלולים מסוגל לנוע ב 8 כוונים שונים והוא מבצע תזוזה אחת לאחר כל מהלך של בולי (לא משנה אם בולי זו או לא באותו מהלך).

אם דוב השבלולים מנסה לבצע תזוזה, אבל הוא נחסם ע"י קיר, הוא פשוט ישאר במקומו עד התור הבא. כיצד נדע איך להזיז את הדוב?

נבצע בנפרד חישוב לתזוזה בטור ולתזוזה בשורה.

כדי למצוא מה התזוזה בטור נבצע חיסור בין הטור שבו נמצא בולי לטור שבו נמצא הדוב .

אם התוצאה היא שלילית תזוזת הדב תהיה - 1 (שמאלה לכיוון של בולי).

אם התוצאה היא חיובית התזוזה בטור +1 (ימינה לכיוון של בולי).

אם התוצאה היא 0 לא תהיה תזוזה באותו הטור.

חישוב זה נבצע לגבי התזוזה בשורה.

דוגמא:

1	2	3	4
*	*	*	D
#	*	#	#
#	B	#	*
*	*	*	*

מקרא:



* - קיר מבוכ

- נקודה שצריך לאסוף.

B - בולי

D - דוב השבלולים

הדוב נמצא ב – (1,4) ובולי נמצא ב (3,2)

ההפרש בין הטורים הוא (2-) כלומר התזוזה בטור תהיה (1-) (הטור של הדוב גדול משל בולי ולכן נקטין אותו).

ההפרש בין השורות הוא 2 כלומר התזוזה בשורה תהיה (1+) (השורה של הדוב קטנה מהשורה של בולי ולכן

נגדיל אותה).

כלומר לאחר המהלך הנוכחי מיקומו של הדוב יהיה: (2,3)

הפלט:

צורת הסיום של המשחק:

האם הצליח בולי לאכול את כל הנקודות או שדוב השבלולים הצליח לתפוס אותו ולבצע בו את זממו.

הערות ודגשים מיוחדים:

מובטח שהקלט תקיין (למעט הדברים עליהם נאמר במפורש שצריך לבדוק).

מובטח שמקומו ההתחלתי של בולי אינו על קיר מבוכ. מה זה אומר ? - שהוא נמצא על #. את ה - # הזה הוא

אמור לאכול!

כשדוב השבלולים נע במבוכ הוא אינו אוכל אתה - # (הוא אוהב רק שבלולים).

בתחילת המשחק, עד שבולי לא מתחיל לזוז גם דוב השבלולים לא זז.

יש לדאוג שבולי לא יעוף מהלוח (והלוח אינו ציקלי....)



דוגמא שלישית - מילון אוקספורד

חברת אוקספורד החליטה שהיא צריכה להתחיל לכתוב מילון חדש. המנהל של אוקספורד פרופסור אוק ספורד, מחפש כוח – עבודה אשר יפיק את המרב מכתובת המילון. והכוח עבודה הנבחר? **חניכי קורס תכנות! מה מאפשר המילון?**

תפריט המילון נותן למשתמש מספר אופציות:

א. חיפוש הגדרה של מילה, ע"פ המילה.

דוגמא: המשתמש מקיש, "Computer" ומקבל חזרה פלט למסך:

A_machine_that_calculates_diffrent_calculations

ב. הכנסת מילה חדשה והגדרתה למילון.

דוגמא: המשתמש מקיש את המילה Computer ולאחריה המשתמש יתבקש לתת את ההגדרה

למילה. מובטח כי בהגדרת המילה, המשתמש ישתמש ב() קו תחתון ולא בתווי רווח (" ").

כעת, המילה נוספה למילון, ויהיה ניתן תמיד לחפש אותה ואת הגדרתה.

ג. הדפסת כל המילים במילון אשר מתחילות באות מסוימת.

דוגמא : המשתמש מקיש את האות C, יודפסו כל המילים אשר קיימות במילון, ללא הגדרתן,

המתחילות באות C.

ד. יציאה מהתכנית.

איך המילון ימומש?

המילון ימומש באמצעות קובץ טקסט. שם הקובץ הוא " Dictionary.dat ". הקובץ ימוקם תמיד בספריית הפרוייקט.

מבנה הקובץ:

בכל שורה בקובץ ישנה מילה, אשר היא המילה לחיפוש. לאחר המילה מופיע תו רווח, ואחריו הגדרת המילה.

לדוגמא :

Computer A_machine_that_calculates_diffrent_calculations

Sound What_the_ear_hears.

על מנת לממש את המילון עלינו להשתמש בשגרות, במצביעים, ובהקצאה דינמית.

להלן פירוט השגרות בהן נשתמש:

הוספת מילה, הגדרת השגרה:

`void AddNewWord(char* iszNewWord, char* iszWordDef)`

שגרה זו מוסיפה מילה לקובץ המילון.

קבלת הגדרת מילה, הגדרת השגרה:

<http://cms.education.gov.il/EducationCMS/UNITS/MadaTech/csit>



char* GetWordDefenition(**char*** iszWord)

שגרה זו מקבלת מילה ומחזירה מחרוזת המכילה שרשור של המילה וההגדרה של המילה מהמילון. המילה וההגדרה יופרדו באמצעות- .

למשל: במידה ושלחנו את המילה sound תוחזר המחרוזת:

Sound-What_the_ear_hears

***במידה והמילה אינה נמצאת בקובץ יש להחזיר מצביע NULL**

על מנת לשרשר בין המילה וההגדרה יש להשתמש בשגרה שאתם תממשו:

void JoinStrings(char iopszFirstString, char* iszSecondString)**

הדפסת כל המילים המתחילים באות מסוימת, הגדרת השגרה:

void PrintWordsOf(char icStartChar)

שגרה זו מדפיסה את כל המילים המתחילים באות מסוימת, ללא הגדרתן. אלגוריתם הפעלה לשיקול התכניתן- . שגרה ראשית, הגדרת השגרה:

void main()

שגרה זו מנהלת את התוכנית, קוראת לשגרות המתאימות. יש להדפיס את התפריט למשתמש כל עוד המשתמש לא בחר לצאת מהתוכנית.

דברים נוספים:

בדיקות תקינות:

1. מובטח שקליטת המילים מהמשתמש תקינה.

2. יש לבדוק את קוד הפעולה שהמשתמש מקיש.

אורך חיי המשתנים:

1. יש להשתמש בהקצאה דינמית בשגרה המתאימה, על מנת לצמצם את הזיכרון בשימוש.

2. יש לזכור לשחרר את הזיכרון כאשר צריך.

3. אין להגדיר את הקובץ כגלובלי, מכיוון שאנו מבצעים את פעולת הפתיחה בכל פעם מחדש,

4. לאורך כל התכנית מובטח כי הגודל המקסימלי של מילה הוא 20 תווים, והגודל המקסימלי של הגדרה הינו 200 תווים. מכיוון שאנו רוצים לחסוך בזיכרון אשר יוקצה לתוכנית הראשית, אנו נרצה להקצות באופן דינמי את הגודל המינימלי של תווים אשר עלינו לשלוח חזרה לתוכנית.



גמישות עיצוב המערכת:

בעיצוב מערכת זו ניתן להוסיף שגרות כאוות נפשכם. כדאי לחקור ולהוסיף פיצ'רים למילון!



דוגמא רביעית – משחק טנקים

כללי

עליכם לכתוב תוכנית אשר תדמה משחק האסטרטגיה של שני טנקים.

עקרון המשחק

המשחק הולך כך: על לוח המשחק ממוקמים שני טנקים (עבור שני משתמשים), כאשר כל טנק משחק בתורו. על המשתמשים לבחור את סוג הפעולה אותה רוצה לבצע על מנת להצליח במטרה סופית לפגוע בטנק היריב.

איך עושים את זה??

יש לנו לוח משחק בגודל 11 X 11 משבצות (ניתן להסתמך על כך שאורך ורוחב הלוח יהיה תמיד אי זוגי).

	X		X		X		X		X	
	X		X		X		X		X	
O	X		X		X		X		X	O
	X		X		X		X		X	
	X		X		X		X		X	

מקרא:

X חומה שבירה (תהיה בתאים בהם הטור והשורה זוגיים)

O שני הטנקים (אחד מכל צד בקצוות הלוח באמצע הטור)

המשחק מתנהל בתורות כאשר הטנק השמאלי מתחיל, ואחריו הימני וחוזר חלילה.

בכל תור לטנק יש 4 אפשרויות:

1. פקודת תנועה - בפקודה זו המשתמש יזין כיוון תנועה ואת מס' משבצות שהוא רוצה לנוע. הטנק יזוז את

מספר המשבצות המבוקש, אלא אם כן הגיע לפני כן לחומה, לטנק השני או אחד מגבולות הלוח. במקרה

זה, ייעצר הטנק ויגיע עד לנקודת ההיתקלות.

2. פקודת ירי רגיל - בפקודה זו המשתמש יזין כיוון ירי. במידה והירייה הגיעה לחומה שבירה היא תהרוס

אותה ורק אותה (לא תמשיך הלאה), במידה והירייה הגיעה לקצה הלוח לא יקרה כלום, ובמידה והירייה

הגיעה אל הטנק השני המשתמש שביצע את הירי ניצח.

<http://cms.education.gov.il/EducationCMS/UNITS/MadaTech/csit>



3. ירי בכיוון מתחלף - פקודה זו דומה לירי רגיל. המשתמש מזין כיוון ירי **התחלתי**. במידה והירייה הגיעה לחומה שבירה היא תהרוס אותה ורק אותה (לא תמשיך הלאה), ובמידה והירייה הגיעה אל טנק תפגע בו ותגרום לסיום המשחק. במידה והירייה הגיעה לקצה הלוח, **בניגוד לירי רגיל**, הירייה **תשנה את כיוונה** ימינה מהכיוון הקודם ותמשיך ישר עד הגעתה לחומה שבירה (בה תעצר) או טנק (תהרוס אותו ותגרום לסיום המשחק), או קצה הלוח (תשנה את כיוונה שוב).
- יש **לשים לב** לשלושה דברים בפקודה זו:
- א. הירייה תסיים את פעולתה ברגע שתגיע לחומה שבירה, או לטנק.
- ב. הירייה מסוכנת כי היא **מסוגלת לפגוע גם בטנק שביצע את הירי מלכתחילה**, כלומר ירי עצמי ובכך לגרום למשתמש שביצע את הירייה להפסיד.
- ג. במידה והירייה לא פגעה בטנק או בחומר שבירה, היא תיפסק לאחר שינויי כיוון (כדי שלא יקרה מצב שהיא ממשיכה לנוע ללא הפסקה).
4. פקודת הטלת פצצה אווירית - בפקודה זו המשתמש מטיל פצצה **רנדומלית** במיקום בו יבחר. גודל הפצצה האווירית היא תמיד בגודל 3×3 משבצות, כאשר חלק מהתאים בה דלוקים ולחלק כבויים. למשל:

0	1	1
0	0	0
1	0	1

- עבור כל אחת מהמשבצות נבצע פקודת Random שתגריל מספר בין 0 ל-1 שייקבע האם התא דלוק או כבוי. כל משבצת דלוקה (מכילה את המספר 1) תחסל את מה שיש בלוח המשחק במקום בו פגעה (חומה שבירה או את הטנק השני), ואילו משבצת כבויה (מכילה את המספר אפס) לא תבצע דבר למשבצת שתחתיה. הפצצה תמוקם במיקום בו יבחר המשתמש, כאשר המיקום מייצג את מיקום המשבצת **האמצעית** של הפצצה האוויר (כלומר תא 1,1 בפצצה).
- יש **לשים לב** לשני דברים בפקודה זו:
- במידה והפצצה האווירית יוצאת מגבולות לוח המשחק, היא תהיה אפקטיבית רק לתאים שעל תחום הלוח.** כמו בפקודה הקודמת, גם כאן הטלת הפצצה מסוכנת גם לטנק היורה שכן הוא לא חסין אליה ובמקרה של הטלת הפצצה מעל הטנק היורה יתכן מצב שיפגע בעצמו.
- המשחק נגמר ב 2 מצבים:



1. אחד מהטנקים נפגע(בין אם פגע בעצמו או נפגע ע"י הטנק השני) במקרה זה הטנק השורד הוא המנצח.
2. הלוח ריק מחומות שבירות במקרה זה תוצאה המשחק היא תיקו- .
3. כאשר המשתמש מכניס את המספר 999 בתפריט המשחק, ריצת המערכת תסתיים.

דגשים:

- הכיוונים במשחק הם : 1- למעלה, 2- ימינה , 3- למטה , 4- שמאלה .
- ניתן להסתמך על כך שגודל לוח המשחק יהיה אי זוגי(גם השורות וגם הטורים), אך לא על כך שגודל הלוח הינו 11×11 .
- גירסה נוספת למימוש כאשר גודל הלוח הוא משתנה ואת גודל הלוח קולטים מן הקלט