|  |
| --- |
| **הגדרות ומושגים** |
| **מחשב** - קלט-עיבוד-פלט  **IO** - Input Output  **CPU** - center process unit **Memory** - saving information  שפת תכנות - אוסף של פקודות שבסופו של דבר CPU מתרגם לפעילות  משתנה- קופסא בזיכרון שנתייחס אליה באמצעות השם שלה  תוכנית - אוסף של פקודות שמריץ בזרימה רצף של פקודות באופן סידרתי  **LVALUE** Leftassign  החישוב של התרגיל הולך לצד שמאל ולא לצד ימין  **Case sensitive**  שם משתנה - אותיות באנגלית, בשילוב מספר, התו הראשון יהיה אות  **Build (Compilation)  קומפילציה**  לוקח קובץ טקסט (בן אדם כתב ומכיר) הופך אותו לשפת מכונה. |

|  |
| --- |
| **מבנה לתוכנית בסיס - להעתיק מכאן** |
| #include <stdio.h>  #pragma warning(disable : 4996)  int main()  {      printf("Hi");      return 0;  } |

|  |
| --- |
| זיכרון של המחשב |
| 01000111110101000111101110000101010101010101010001010111111111100010101010  מפה ענקית של ביטים.  Byte = 8 bits = >  2byte = 16bits 4bytes = 32bits  **משתנה -  variable**  משתנה הוא הניגוד של הקבוע  הגדרת משתנה - פקודה שמקצה מקום בזיכרון ונותנת למקום הזה שם. |

ב solution אחד יכולים להיות מספר פרויקטים ובכל פרויקט מספר קבצי C.

אך יש לשים לב שיש רק main אחד. כל פקודה צריכה להסתיים ב;

**פקודות בסיסיות:**

**Printf()** .1

אם אני רוצה להדפיס מספר אני רושםd % במקום שבו אני רוצה שיהיה המספר ולאחר הציטוט " " אני מוסיף את המספר בהתאמה כסדר. מספר ממש או משתנה אינטגרי לחלופין.

דוגמא: printf("I am %d years old and my sister %d", 21,22);

2**. ירידת שורה- n\**  יירשם לרוב בסוף ציטוט. חייב להירשם בתוכו.

דוגמא: printf("another row\n"

3. **scanf()**

אם אני רוצה לקלוט מספר מהמשתמש אני רושם d% יהיה בסוף הציטוט ובתוכו. ניתן לשחרר גם הודעה/הוראה למשתמש. ניתן לקלוט למשתנים רבים אך כולם יקלטו לפי הפורמט שאני יוצר בפקודה. בפקודה עצמה יש לרשום את שם המשתנה שאליו אני רוצה שיקלט המספר ולפניו &. המספרים יקלטו למשתנים בהתאמה לפי הסדר.

דוגמא: scanf("pleas enter 3 numbers %d %d %d", &a, &b, &c);

4.**sizeof(**)

מחזיר את גודל המשתנה בביטים.

5. **תנאי if()**

6. **לולאת while()**

7. **לולאת אינדקס for()**

8. **הערת שורה:** //

9. **הערת מלל:** /\* \*/

10. **d%** מייצר תבנית שאליה יודפס המספר בפלט או יקלט לחלופין. נמצא בתוך פקודות. ניתן להוסיף מספר לפני הd כך שיקבע את גודל התבנית המספר מייצג את מספר הרווחים המוקצים למשתנה שיוצב במקום d.

11.**arr[]** יצירת מערך

12. **s%** תבנית להדפסת מחרוזת

13. **c%** תבנית להדפסת תו

14. **ספריית פונקציות string.h-** בפרק על המחרוזות יש דוגמאות למספר פונקציות.

15. \* פוינטר מצביע

**משתנים וזיכרון**

|  |  |  |
| --- | --- | --- |
| **Type** | **Storage size** | **Value range** |
| Char | 1 byte | -128 to 127 or 0 to 255 |
| unsigned char | 1 byte | 0 to 255 |
| signed char | 1 byte | -128 to 127 |
| Int | 2 or 4 bytes | -32,768 to 32,767 or -2,147,483,648 to 2,147,483,647 |
| unsigned int | 2 or 4 bytes | 0 to 65,535 or 0 to 4,294,967,295 |
| Short | 2 bytes | -32,768 to 32,767 |
| unsigned short | 2 bytes | 0 to 65,535 |
| Long | 8 bytes or (4bytes for 32 bit OS) | -9223372036854775808 to 9223372036854775807 |
| unsigned long | 8 bytes | 0 to 18446744073709551615 |

**כתובת של המשתנה- המקום השמור בזכרון-** &

**תנאים:**

פקודות התוכנית זורמות בצורה לינארית לעתים פקודות מסוימות יבוצעו/ לא יבוצעו לפי תנאים.

התנאים יתבטאו בפונקציית **if()** הביטוי בפונקציה הוא ביטוי בוליאני במידה והביטוי מתבצע והינו אמת הפקודה הנמצאת בתוך הבלוק של if()תבוצע, במידה והביטוי שקר הפקודה בבלוק לא תבוצע ובמידה ויש פקודה בבלוק else פקודה זו תיבצע.

**דוגמאות סוגי תנאים:**

|  |  |  |
| --- | --- | --- |
| Num>100 | Num<100 | Num==100 |
| Num<=100 | Num>=100 | Num!=100 |

**כתיבה מקוצרת להעלאה או הורדה ב1:**

A++/--;

**תנאי מקוצר (דוגמא):**

A=(num>100)? 100:10;

(condition)? (T): (F);

**פעולות חשבון:**

+, -, \*, /, %- מודולו (שארית).

**משתנה עשרוני(לא שלם):** float

**תנאים מורכבים:**

| **שער OR או** |
| --- |
| True || True => True  True || False => True  False || True => True  False || False => False |

|  |
| --- |
| **שער And וגם** |
| True && True => True  True && False => False  False && True => False  False && False => False |

|  |
| --- |
| **שער Not היפוך** |
| !True => False  !False => True  קדימות לסוגרים קדימות ל AND |

|  |
| --- |
| **לולאת תנאי while:** |
| ביצוע קטע קוד פעם אחד או יותר. מורכב מ: תנאי לולאה, מונה לולאה ומקדם מונה.  יש הבדל בין לולאת מונה(שעולה אוטומטית) לבין לולאת תנאי שהתנאי יכול להתקיים/לא להתקיים בהשפעת היוזר(קליטת ערך...). |
| **לולאת forאינדקס (מונה)** |
| לולאת for דומה לולאת while מונה. |

**לולאה מקוננת- לולאה בתוך לולאה**

כל איטרציה חיצונית של הלולאה יתבצעו כל הפקודות שבתוכה עד גמר. גם אם יש 100 איטרציות של לולאה פנימית.

כלומר סך האיטרציות יהיו לולאה חיצונית \* פנימית.

**מערכים**

מערך הוא משתנה היכול להכיל מספר משתנים ממוספרים כמות המשתנים שיכול להכיל יוגדרו בסוגריים המרובעים [ ] בעת יצירת המערך.

מקום מערך יתחיל ממסופר ב-0 ויסתיים במספר המקומות במערך פחות אחת.

למשל: מערך אינטגרי המכיל 30 ציונים של תלמידי הכיתה. הציון הראשון שנקלט יהיה במקום 0 במערך והציון האחרון(30) במקום ה29.

שימוש בלולאות for הכרחי ונחוץ למערכים.

ניתן להשתמש בarr.lenght בלולאות for פקודה הנותנת את אורך המערך.

|  |
| --- |
| **Function פונקציות** |
| 1. Return value 2. Body 3. Parameters 4. Use by call it   פונקציות אשר לא מחזירות ערך נוסיף להן בסוף הפונקציה return 0 ונגדיר את סוג הפונקציה כint בדיוק כמו שפועלת פונקציית main.  יש לשים לב שכאשר מפעילים פונקציה סוג ערך שתחזיר יוגדר כסוג הפונקציה. וסוג הפונקציה יהיה כמו סוג המשתנה שאליו יוחדר הערך כאשר מפעילים את הפוקנציה. |

|  |
| --- |
| **strinמחרוזות** |
| בC **מחרוזת היא מערך של char**  Char str[3]  Str[0]='A'  Str[1]='B'  **מקום בזיכרון**  Char- 1 byte- 8 bits.  יש 256 אפשרויות בchar לפי טבלת ASCII.  ניתן להגדיר משתנים לפי המספר בטבלת אסקי ולפי התו עצמו.  **הגדרת משתנה char:**  Char c=65  Char c2='A'  **על מנת שהמחשב והפונקציות ידעו איכן סוף המחרוזת יש לסמן במספר 0 או null**  Str[2]=0  **הדפסת מחרוזת**  %s 🡪 printf("%s", str)  **הדפסת תו**  c 🡪 printf("%c", str[2])%  יש **ספריית פונקציות** שקשורה למחרוזות שעל מנת שנוכל להפעיל בלי שגיאות עלינו להוסיף למעלה  #include <string.h>  **קליטת מחרוזת-** אין צורך להוסיף כתובת & מכיוון ששם הפונקצייה היא כבר הכתובת של מערך התווים  Gets(str)  מכיוון שאנחנו לא יודעים כמה תווים יכניס המשתמש בקליטה והמשתמש עצמו לא יודע מה גודל המערך עדיף להגדיר תמיד מלכתחילה מערך גדול [1000] ולכן גם לולאת for לא תעבוד בקליטה.  לכן עלינו להשתמש ב**לולאת while** כאשר התנאי הוא כל עוד התו שונה מ0. Str[i]!=0  **אורך מחרוזת**  Strlen()  **העתקת מחרוזת-** מעתיק str2 לstr1  Strcpy(str1, str2)  **שרשור מחרוזות**- הסדר בדומה ללמעלה  Strcat(str1, str2)   * אם משרשרים 2 מחרוזות ושתיהן נקלטות מהמשתמש יש לאתחל את השנייה str2[0]=0   שהפונקציה תדע מאיפה לשרשר  **השוואת פונקציות**- מחזיר 0 אם שוות ומחזיר חיובי או שלילי בהתאמה לאיזו מחרוזת ארוכה יותר  Strcmp(str1, str2)  **בדיקה אם** **פונקציה אחת מוכלת בשנייה**- הסדר בדומה ללמעלה  Strstr(str1, str2)  מחזיר פוינטר מסוג מערך תווים |

|  |
| --- |
| פוינטר מצביע pointer |
|  |

|  |
| --- |
| פוינטרים |
|  |