**pocket calculator project**

**utuyo**



**Ben Marder 200436319**

**Danny Spiegel 301894770**

תרשים התוכנית מאקרו- לאחר כל קלט התוכנית בודקת איזה סוג קלט נכנס לאחר מכן מבצעת פעולה ואז ממתינה לקלט הבא. (למעט Q)

**קלט**

q

(quit)

מספר

כל קלט אחר (שגיאה)

פעולה

אריטמטית

=

תרשים זיהוי קלט:

**קלט**

**כן**

**חישוב עם אופרטור קודם**

**פעולה על המספר**

**סיום תוכנית**

**לא**

**כן**

**לא**

**לא**

**לא**

**לא**

**כן**

**כן**

**כן**

**(+ - / \*)?**

**=?**

**בתחום של האופרטורים?**

**?q**

**מספר?**

**חישוב עם אופרטור קודם ולאחר מכן הדפסה**

**קלט לא נכון**

דוגמא לעבודה של התוכנית כשנרצה לבדוק מה התוצאה של 25\*10 :

1) בתחילת התוכנית num1 ו- num מאותחלים ל-0 ו- operator1 מאותחל ל-+

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שלב | קלט | Operator2 | Operator1 | Num | Num2 | Num1 |
| 1 |  |  | + | 0 |  | 0 |

2) נניח שהכנסו את הסיפרה 2: התוכנית מטפלת בקלט "2" ועושה :

num=multy\_by\_10(num) +(cc – 30);

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שלב | קלט | Operator2 | Operator1 | Num | Num2 | Num1 |
| 1 |  |  | + | 0 |  | 0 |
| 2 | 2 |  | + | 2 |  | 0 |

3) נניח שהכנסו את הסיפרה 5: התוכנית מטפלת בקלט "5" ועושה :

num=multy\_by\_10(num) +(cc – 30);

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שלב | קלט | Operator2 | Operator1 | Num | Num2 | Num1 |
| 1 |  |  | + | 0 |  | 0 |
| 2 | 2 |  | + | 2 |  | 0 |
| 3 | 5 |  | + | 25 |  | 0 |

4) נניח שהכנסנו את האופרטור \* :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שלב | קלט | Operator2 | Operator1 | Num | Num2 | Num1 |
| 1 |  |  | + | 0 |  | 0 |
| 2 | 2 |  | + | 2 |  | 0 |
| 3 | 5 |  | + | 25 |  | 0 |
| 4.א | \* | \* | + | 25 | 25 | 0 |

1. Operator2=\* ו- num2=num
2. מתבצעת פעולת חישוב עם Operator1 , Num1 ו- Num2 כשהתוצאה נכנסת ל- Num1 ( שים לב מתבצע 0+25)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שלב | קלט | Operator2 | Operator1 | Num | Num2 | Num1 |
| 1 |  |  | + | 0 |  | 0 |
| 2 | 2 |  | + | 2 |  | 0 |
| 3 | 5 |  | + | 25 |  | 0 |
| 4.א | \* | \* | + | 25 | 25 | 0 |
| 4.ב |  | \* | + | 25 | 25 | 25 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שלב | קלט | Operator2 | Operator1 | Num | Num2 | Num1 |
| 1 |  |  | + | 0 |  | 0 |
| 2 | 2 |  | + | 2 |  | 0 |
| 3 | 5 |  | + | 25 |  | 0 |
| 4.א | \* | \* | + | 25 | 25 | 0 |
| 4.ב |  | \* | + | 25 | 25 | 25 |
| 4.ג |  | \* | \* | 0 | 25 | 25 |

1. Operator2= Operator1 , num=0

5) נניח שהכנסו את הסיפרה 1: התוכנית מטפלת בקלט "1" ועושה :

num=multy\_by\_10(num) +(cc – 30);

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שלב | קלט | Operator2 | Operator1 | Num | Num2 | Num1 |
| 1 |  |  | + | 0 |  | 0 |
| 2 | 2 |  | + | 2 |  | 0 |
| 3 | 5 |  | + | 25 |  | 0 |
| 4.א | \* | \* | + | 25 | 25 | 0 |
| 4.ב |  | \* | + | 25 | 25 | 25 |
| 4.ג |  | \* | \* | 0 | 25 | 25 |
| 5 | 1 | \* | \* | 1 | 25 | 25 |

6) נניח שהכנסנו את הסיפרה 0: התוכנית מטפלת בקלט "0" ועושה :

num=multy\_by\_10(num) +(cc – 30);

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שלב | קלט | Operator2 | Operator1 | Num | Num2 | Num1 |
| 1 |  |  | + | 0 |  | 0 |
| 2 | 2 |  | + | 2 |  | 0 |
| 3 | 5 |  | + | 25 |  | 0 |
| 4.א | \* | \* | + | 25 | 25 | 0 |
| 4.ב |  | \* | + | 25 | 25 | 25 |
| 4.ג |  | \* | \* | 0 | 25 | 25 |
| 5 | 1 | \* | \* | 1 | 25 | 25 |
| 6 | 0 | \* | \* | 10 | 25 | 25 |

7) נניח שהכנסנו את האופרטור = :

1. num2=num

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שלב | קלט | Operator2 | Operator1 | Num | Num2 | Num1 |
| 1 |  |  | + | 0 |  | 0 |
| 2 | 2 |  | + | 2 |  | 0 |
| 3 | 5 |  | + | 25 |  | 0 |
| 4.א | \* | \* | + | 25 | 25 | 0 |
| 4.ב |  | \* | + | 25 | 25 | 25 |
| 4.ג |  | \* | \* | 0 | 25 | 25 |
| 5 | 1 | \* | \* | 1 | 25 | 25 |
| 6 | 0 | \* | \* | 10 | 25 | 25 |
| 7.א | = | \* | \* | 10 | 10 | 25 |

1. מתבצעת פעולת חישוב עם Operator1 , Num1 ו- Num2 כשהתוצאה נכנסת ל- Num1 (25\*10)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שלב | קלט | Operator2 | Operator1 | Num | Num2 | Num1 |
| 1 |  |  | + | 0 |  | 0 |
| 2 | 2 |  | + | 2 |  | 0 |
| 3 | 5 |  | + | 25 |  | 0 |
| 4.א | \* | \* | + | 25 | 25 | 0 |
| 4.ב |  | \* | + | 25 | 25 | 25 |
| 4.ג |  | \* | \* | 0 | 25 | 25 |
| 5 | 1 | \* | \* | 1 | 25 | 25 |
| 6 | 0 | \* | \* | 10 | 25 | 25 |
| 7.א | = | \* | \* | 10 | 10 | 25 |
| 7.ב | = | \* | \* | 10 | 10 | 250 |

1. התוכנית מדפיסה את Num1 ומאפסת את Num1 Num, ומאתחלת את Operator1 ל-+ .

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| שלב | קלט | Operator2 | Operator1 | Num | Num2 | Num1 | print |
| 1 |  |  | + | 0 |  | 0 |  |
| 2 | 2 |  | + | 2 |  | 0 |  |
| 3 | 5 |  | + | 25 |  | 0 |  |
| 4.א | \* | \* | + | 25 | 25 | 0 |  |
| 4.ב |  | \* | + | 25 | 25 | 25 |  |
| 4.ג |  | \* | \* | 0 | 25 | 25 |  |
| 5 | 1 | \* | \* | 1 | 25 | 25 |  |
| 6 | 0 | \* | \* | 10 | 25 | 25 |  |
| 7.א | = | \* | \* | 10 | 10 | 25 |  |
| 7.ב |  | \* | \* | 10 | 10 | 250 |  |
| 7.ג |  | \* | + | 0 |  | 0 | 250 |

הערות:

1. התוכנית פעולת בשיטת "busy waiting"
2. התכנית יודעת לחשב מספר פעולות חשבוניות ורק לאחר קלט של "=" תדפיס את התוצאה
3. קלט של "=" מחזיר אותך לתחילת התוכנית
4. פונקציית הכפל פועלת ככפל בינארי
5. התכנית יודעת לטפל במקרה של הכנסת מספר לחיצה על פעולה חשבונית ולאחר מכן לחיצה על שווה ( =+6 לדוגמה)
6. לפעולות החיבור והחיסור ולקלט יש בדיקת overflow underflow
7. התכנית משתמשת פעמיים ב-"flag" :
8. במקרה של הכנסת שני קלטים ברצף של אופרטור(על מנת לדרוס אופרטור קודם) ולמקרה של הכנסת קלט של "=" לאחר קלט של אופרטור
9. בפונקציית החילוק הפעולה מתבצעת בערך מוחלט והדגל מופעל מקרה שהתוצאה אמורה להיות שלילית