1) מצורף בקובץ py בשם q1.
2) מצורף מטה BNF:

a. <program> ::= <statement> | <statement> "\n" <program>
b. <statement> ::= <assignment> | <if_statement>
c. <assignment> ::= <variable> "=" <expression>
d. <if_statement> ::= "if" <expression> ":" <statement> ":" <statement>
e. <expression> ::= <term> | <expression> "+" <term> | <expression> "-" <term>
f. <term> ::= <factor> | <term> "*" <factor> | <term> "//" <factor>
g. <factor> ::= <number> | <variable> | "(" <expression> ")"
h. <variable> ::= <letter> | <variable> <letter>
i. <number> ::= <digit> | <number> <digit>
j. <letter> ::= "a" | "b" | "c" | ... | "z" | "A" | "B" | "C" | ... | "Z" | "_"
k. <digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

3) מצורף בקובץ py בשם q3. ה-BNF נשאר זהה לסעיף הקודם.
4) השפה שלנו מוגדרת באופן דינאמי. בשפה דינאמית סוגי המשתנים נקבעים בזמן ריצה וניתן להקצות מחדש משתנים לסוגים שונים של ערכים. זאת בניגוד לשפות שמוגדרות באופן סטטי שבהן יש להצהיר על סוגי משתנים בזמן הקומפליציה והם לא יכולים להשתנות.
5) מצורף בקובץ py בשם q5q6. מצורף מטה BNF עבור הסעיף:

a. <program> ::= <statement> | <statement> "\n" <program>
b. <statement> ::= <assignment> | <if_statement>
c. <assignment> ::= <variable> "=" <expression>
d. <if_statement> ::= "if" <expression> ":" <statement> ":" <statement>
e. <expression> ::= <term> | <expression> "+" <term> | <expression> "-" <term>
f. <term> ::= <factor> | <term> "*" <factor> | <term> "//" <factor>
g. <factor> ::= <number> | <variable> | "(" <expression> ")"
h. <variable> ::= <letter> | <variable> <letter>
i. <number> ::= <digit> | <number> <digit>
j. <letter> ::= "a" | "b" | "c" | ... | "z" | "A" | "B" | "C" | ... | "Z" | "_"
k. <digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
l. <expression> ::= <comparison> | <expression> "+" <comparison> | <expression> "-" <comparison>
m. <comparison> ::= <term> | <comparison> ">" <term> | <comparison> "<" <term> | <comparison> "==" <term>

6) מצורף בקובץ py בשם q5q6. ה-BNF נשאר זהה לסעיף הקודם.
7) מצורף בקובץ py בשם q7. מצורף מטה BNF עבור הסעיף:

a. <program> ::= <statement> | <statement> "\n" <program>
b. <statement> ::= <assignment> | <if_statement>
c. <assignment> ::= <variable> "=" <expression>
d. <if_statement> ::= "if" <expression> ":" <statement> ":" <statement>
e. <expression> ::= <term> | <expression> "+" <term> | <expression> "-" <term>
f. <term> ::= <factor> | <term> "*" <factor> | <term> "//" <factor>
g. <factor> ::= <number> | <variable> | "(" <expression> ")"
h. <variable> ::= <letter> | <variable> <letter>

i. <number> ::= <digit> | <number> <digit>
j. <letter> ::= "a" | "b" | "c" | ... | "z" | "A" | "B" | "C" | ... | "Z" | "_"
k. <digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
l. <expression> ::= <comparison> | <expression> "+" <comparison> | <expression> "-" <comparison>
m. <comparison> ::= <term> | <comparison> ">" <term> | <comparison> "<" <term> | <comparison> "==" <term>
n. <statement> ::= <assignment> | <if_statement> | <while_statement>
o. <while_statement> ::= "while" <expression> ":" <program>

8) מצורף בקובץ py בשם q8.
9) מצורף בקובץ py בשם q9.
10) מצורף בקובץ py בשם q10.
11) מצורף בקובץ py בשם q11.
12) מצורף בקובץ py בשם q12.
13) מצורף בקובץ py בשם q13.
14) ההבדלים בין שני קטעי הקוד הוא שהקטע ה"עצל" קורא ללואת ה-FOR באופן ישיר מבלי לקרוא לרשימה של generate_values , כלומר החישוב שלה לא נעשה באופן מיידי אלא לפי הצורך כאשר בכל פעם נוצר/נלקח אלמנט אחד לביצוע הפעולה ככה שהזיכרון לא מוקצה מראש אלא רק בעת החישוב.
לעומת זאת בקטע ה"נלהב" כל הערכים של הרשימה נוצרים באופן מיידי ומוקצה עבורם ערך וזיכרון, רק לאחר מכן עוברים על כל המערך ומבצעים את פעולת החזקה.