תיק מלווה פרוייקט באסמבלי לקראת בחינת הבגרות 899283

נושא העבודה: קליטת משפטים

שם התלמיד: דביר ברלוביץ

בית ספר: ישיבת בני עקיבא גבעת שמואל

כיתה: י'2

מקצוע: ארגון המחשב וספת סף

תעריך הגשה: י' אייר תש"פ (4.5.2020)

סביבת פיתוח: EMU8086

מורה: גמליאל זאבי

תוכן עיניינים

[**מבוא קצר** 3](#_Toc39397998)

[**תרשים זרימה של התוכנית** 4](#_Toc39397999)

[**תיאור מהלך העבודה בליווי דוגמאות** 5](#_Toc39398000)

[**פירוט השגרות של התוכנית** 7](#_Toc39398001)

[**תיעוד משתנים** 9](#_Toc39398002)

[**הקוד** 11](#_Toc39398003)

# **מבוא קצר**

משתמש יקר שלום,

תוכנה זו קולטת קטע טקסט באנגלית באותיות גדולות וקטנות.

בקטע יכולים להופיע הרבה משפטים, כל משפט מסתיים בנקודה.

אורך הקטע עד 200 תווים.

בסיום הזנת המשפט התוכנית תציג:

1. את הקטע כשהוא כתוב כולו באותיות קטנות, מלבד האותיות הגדולות בתחילת משפט.
2. את כל המשפטים בקטע מן הארוך ביותר עד הקצר ביותר.
3. את מספר המשפטים בקטע.
4. את מספר המילים בקטע.

# **תרשים זרימה של התוכנית**

התחלה

כן

קלט לא תקין

שאילת: האם לקלוט עוד משפט?

הדפסת המשפטים על פי הסדר

סידור המשפטים לפי אורכם מהארוך לקצר

לא

**סיום**

הדפסת מספר המשפטים בקטע

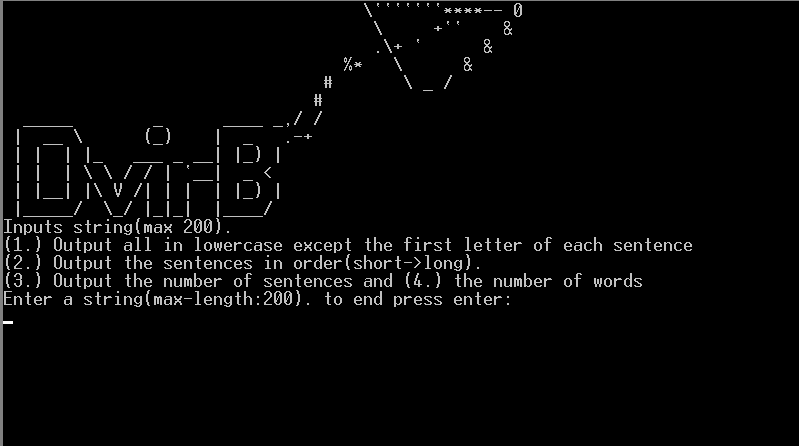
הדפסת מספר המילים בקטע

הדפסת הקטע באותיות קטנות, מלבד האותיות הגדולות בתחילת משפט

קליטת הטקסט עד 200 תווים

הצגת לוגו ותיאור

# **תיאור מהלך העבודה בליווי דוגמאות**



בתחילה התוכנית יוצג הלוגו, תיאור התוכנית ותתחיל קליטה של קטע טקסט.

לאחר מכן, על המשתמש להזין קטע טקסט(עד 200 תווים). לדוגמא:



מיד לאחר שהמשתמש יסיים להזין את הטקסט התוכנית תעבד את הקטע ותציג:

את הקטע באותיות קטנות, מלבד האותיות הגדולות בתחילת משפט,

את הקטע מסודר מהמשפט הארוך לקצר,

את מספר המשפטים ואת מספר המילים.



לבסוף התוכנית תשאל את המשתמש האם לחזור ולקלוט קטע טקסט נוסף או לצאת מן התוכנית.

# **פירוט השגרות של התוכנית**

|  |  |
| --- | --- |
| **תת- תכנית** | **תפקיד, טענת כניסה וטענת יציאה** |
| Main | **תפקיד:** ריכוז כל שגרות התוכנית בתוך שגרה ראשית.  **טענת כניסה:** אין.  **טענת יציאה:** אין. |
| Begin | **תפקיד:** הצגת הלוגו ותיאור התוכנית ואיפוס המשתנים.  **טענת כניסה:** לוגו ותיאור התוכנית.  **טענת יציאה:** איפוס משתנים והצגת לוגו. |
| Input | **תפקיד:** קליטה ועיבוד הטקסט  **טענת כניסה:** הודעת פתיחה לקליטה.  **טענת יציאה:** שמירת הטקסט המעובד במחרוזת. |
| OutputP | **תפקיד:** הצגת הקטע באותיות קטנות מלבד האותיות הגדולות בתחילת משפט.  **טענת כניסה:** מחרוזת הקטע מעובדת.  **טענת יציאה:** הצגת המחרוזת המעובדת. |
| saveLen | **תפקיד:** שמירת מצביע למשפט(נק' ההתחלה של המשפט בקטע ואורך המשפט).  **טענת כניסה:** נק' ההתחלה של המשפט בקטע ואורך המשפט.  **טענת יציאה:** שמירה במערך המצביעים. |
| Sort | **תפקיד:** סידור מערך המצביעים מהארוך לקצר.  **טענת כניסה:** מערך מצביעים ומספר המשפטים בקטע.  **טענת יציאה:** שמירה של המערך כשהוא מסודר. |
| OutputOrder | **תפקיד:** הצגת הקטע על פי הסדר(מהארוך לקצר).  הדפסה בדילוגים על פי מערך המצביעים:  קופצים לנקודת ההתחלה של המשפט הראשון במערך המצביעים ומדפיסים עד סופו על פי האורך וכך ממשיכים למשפטים הבאים על פי המערך.  **טענת כניסה:** מערך מצביעים ומחרוזת הקטע  **טענת יציאה:** הצגת הקטע מסודר. |
| OutputS | **תפקיד:** הצגת מספר המשפטים בקטע.  **טענת כניסה:** מספר המשפטים בקטע.  **טענת יציאה:** הצגת מספר המשפטים בקטע. |
| OutputW | **תפקיד:** הצגת מספר המילים בקטע.  **טענת כניסה:** מספר המילים בקטע.  **טענת יציאה:** הצגת מספר המילים בקטע. |
| Repeat | **תפקיד:** שאלה האם לחזור ולקלוט עוד משפט לעיבוד?  **טענת כניסה:** הודעת השאלה  **טענת יציאה:** אין. |
| exit | **תפקיד:** יציאה מהתוכנית בלחיצת תו כלשהוא.  **טענת כניסה:** הודעת יציאה  **טענת יציאה:** אין. |

# **תיעוד משתנים**

|  |  |  |
| --- | --- | --- |
| **משתנה** | **טיפוס נתונים(בית/ מילה/ מערך/ מחרוזת)** | **תפקיד** |
| logo | מחרוזת | הודעה המציגה את הלוגו |
| exitMsg | מחרוזת | הודעת יציאה |
| descrMsg | מחרוזת | הודעת תיאור התוכנית |
| iMsg | מחרוזת | הודעה למשתמש להקליד קטע טקסט |
| repeatMsg | מחרוזת | הודעה המציגה שאלה האם לחזור ולקלוט עוד קטע |
| invalidMsg | מחרוזת | הודעה המציגה שהקלט אינו תקין |
| itxtLN | מחרוזת | הודעה של קפיצת שורה |
| Itxt | מחרוזת | מחרוזת ריקה שבה קטע הטקסט יישמר |
| order | מערך | מערך המצביעים(אינדקסים) של המשפטים ואורכם |
| senMsg | מחרוזת | הודעת פתיחה להצגת מספר המשפטים |
| sen100 | בית | הצגת ספרת המאות של מספר המשפטים |
| sen10 | בית | הצגת ספרת העשרות של מספר המשפטים |
| sen1 | בית | הצגת ספרת היחידות של מספר המשפטים |
| wordsMsg | מחרוזת | הודעת פתיחה להצגת מספר המילים |
| words100 | בית | הצגת ספרת המאות של מספר המילים |
| words10 | בית | הצגת ספרת העשרות של מספר המילים |
| words1 | בית | הצגת ספרת היחידות של מספר המילים |
| sens | בית | מונה משפטים |
| words | בית | מונה מילים |
| len | בית | מונה זמני לאורך המשפט לצורך שמירה במערך המצביעים |
| point | בית | שמירה זמנית של נקודת ההתחלה של משפט בקטע לצורך שמירה במערך המצביעים |
| idot | בית | ערך(בוליאני 0/1) האם התו האחרון שהוזן הוא נקודה |
| diff | בית | הפרש ascii בין אותיות קטנות לגדולות |
| count | בית | מונה בלולאת קליטת קטע טקסט תו-תו |
| saves | בית | מונה שמירות במערך המצביעים |
| counter | מילה | מונה בלולאה של הדפסת הקטע על פי מערך המצביעים |

# **הקוד**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 |  | .MODEL small |  |
| 2 |  | .STACK 100h |  |
| 3 |  | .DATA |  |
| 4 | ;logo: |  | ;logo: |
| 5 |  | logo db ' \```````\*\*\*\*-- 0 ',13,10 |  |
| 6 |  | db ' \ +`` & ',13,10 |  |
| 7 |  | db ' .\+ ` & ',13,10 |  |
| 8 |  | db ' %\* \ & ',13,10 |  |
| 9 |  | db ' # \ \_ / ',13,10 |  |
| 10 |  | db ' # ',13,10 |  |
| 11 |  | db ' \_\_\_\_\_ \_ \_\_\_\_ \_,/ / ',13,10 |  |
| 12 |  | db ' | \_\_ \ (\_) | \_ .-+ ',13,10 |  |
| 13 |  | db ' | | | |\_ \_\_\_ \_ \_\_| |\_) | ',13,10 |  |
| 14 |  | db ' | | | \ \ / / | `\_\_| \_ < ',13,10 |  |
| 15 |  | db ' | |\_\_| |\ V /| | | | |\_) | ',13,10 |  |
| 16 |  | db ' |\_\_\_\_\_/ \\_/ |\_|\_| |\_\_\_\_/ ',13,10,'$' |  |
| 17 |  |  |  |
| 18 |  |  |  |
| 19 |  |  |  |
| 20 | ; --------------- vars: --------------- | | |
| 21 |  |  |  |
| 22 |  | descrMsg db 'Inputs string(max 200).',13,10 |  |
| 23 |  | db '(1.) Output all in lowercase except the first letter of each sentence',13,10 |  |
| 24 |  | db '(2.) Output the sentences in order(short->long).',13,10 |  |
| 25 |  | db '(3.) Output the number of sentences and (4.) the number of words',13,10,'$' |  |
| 26 |  | 200). to end press enter:',13,10,'$' | iMsg db 'Enter a string(max-length: |
| 27 |  | repeatMsg db 13,10,'Repeat? (y/n)$' |  |
| 28 |  | invalidMsg db 'Input invalid.',13,10,'$' |  |
| 29 |  |  |  |
| 30 |  | itxtLN db 13,10 |  |
| 31 |  | itxt db 200 dup(0) |  |
| 32 |  | db 13,10,'$' |  |
| 33 |  | order db 200 dup(0) |  |
| 34 |  |  |  |
| 35 |  | ' | senMsg db 'sentences: |
| 36 |  | sen100 db 0 |  |
| 37 |  | sen10 db 0 |  |
| 38 |  | sen1 db 0 |  |
| 39 |  | db 13,10,'$' |  |
| 40 |  | ' | wordsMsg db 'words: |
| 41 |  | words100 db 0 |  |
| 42 |  | words10 db 0 |  |
| 43 |  | words1 db 0 |  |
| 44 |  | db 13,10,'$' |  |
| 45 |  |  |  |
| 46 |  | sens db 0 |  |
| 47 |  | words db 0 |  |
| 48 |  | len db 0 |  |
| 49 |  | point db 0 |  |
| 50 |  | idot db 1 |  |
| 51 |  | diff db 0 |  |
| 52 |  | count db 0 |  |
| 53 |  | saves db 0 |  |
| 54 |  |  |  |
| 55 |  | counter dw 0 |  |
| 56 |  | .CODE |  |
| 57 | ;DS must be here. else- JMP MAIN wouldn't work. |  |  |
| 58 |  | mov AX,@data |  |
| 59 |  | mov DS,AX |  |
| 60 |  |  |  |
| 61 |  |  |  |
| 62 |  | ; ------------------------------------------------------------ ; | |
| 63 |  | main: ; main routine ; | main: ; main routine ; |
| 64 |  | ; ------------------------------------------------------------ ; | |
| 65 |  | call input |  |
| 66 |  | call outputP |  |
| 67 |  | call sort |  |
| 68 |  | call outputOrder |  |
| 69 | ;new line: |  | ;new line: |
| 70 |  | mov AH, 02h |  |
| 71 |  | mov DL, 13 |  |
| 72 |  | int 21h |  |
| 73 |  | mov DL, 10 |  |
| 74 |  | int 21h |  |
| 75 |  |  |  |
| 76 |  | call outputS |  |
| 77 |  | call outputW |  |
| 78 |  | call repeat |  |
| 79 |  | jmp exit |  |
| 80 |  |  |  |
| 81 |  |  |  |
| 82 | ; ------------------------------------------------------------ ; | | |
| 83 | begin: ; logo, description, diff and zeroing | | |
| 84 | ; ------------------------------------------------------------ ; | | |
| 85 |  |  |  |
| 86 | ;output logo: |  | ;output logo: |
| 87 |  | mov AH, 09h |  |
| 88 |  | lea DX, logo |  |
| 89 |  | int 21h |  |
| 90 | ;output description of project: |  | ;output description of project: |
| 91 |  | mov AH, 09h |  |
| 92 |  | lea DX, descrMsg |  |
| 93 |  | int 21h |  |
| 94 |  |  |  |
| 95 | ;set diff |  |  |
| 96 |  | mov diff, 'a' |  |
| 97 |  | sub diff, 'A' |  |
| 98 |  |  |  |
| 99 | ;zero |  |  |
| 100 |  | mov sens, 0 |  |
| 101 |  | mov words, 0 |  |
| 102 |  | mov len, 0 |  |
| 103 |  | mov point, 0 |  |
| 104 |  | mov idot, 1 |  |
| 105 |  | mov count, 0 |  |
| 106 |  | mov saves, 0 |  |
| 107 |  | mov counter, 0 |  |
| 108 |  |  |  |
| 109 | ;zero all itxt |  |  |
| 110 |  | mov AX, 0 |  |
| 111 |  |  | zero: |
| 112 |  | add BX, AX |  |
| 113 |  | lea BX, itxt |  |
| 114 |  | mov [BX], 0 |  |
| 115 |  | inc AX |  |
| 116 |  | cmp AX, 200 |  |
| 117 |  | je retu0 |  |
| 118 |  | jmp zero |  |
| 119 |  |  |  |
| 120 |  |  | retu0: |
| 121 |  | ret |  |
| 122 |  |  |  |
| 123 | ; ------------------------------------------------------------ ; | | |
| 124 | input: ; input the string, char by char and process it. ; | | |
| 125 | ; ------------------------------------------------------------ ; | | |
| 126 | ;iMsg |  |  |
| 127 |  | mov AH, 09h |  |
| 128 |  | lea DX, iMsg |  |
| 129 |  | int 21h |  |
| 130 |  |  |  |
| 131 | ;counter | mov count, 0 |  |
| 132 |  | mov idot, 1 |  |
| 133 |  |  | char: |
| 134 |  | mov AH, 01h |  |
| 135 |  | int 21h |  |
| 136 |  |  |  |
| 137 | ;enter |  |  |
| 138 |  |  |  |
| 139 |  | je enter |  |
| 140 |  |  |  |
| 141 |  | inc len |  |
| 142 | ;computer array index at BX |  |  |
| 143 |  | lea BX, itxt |  |
| 144 |  | mov CL, count |  |
| 145 |  | mov CH, 0 |  |
| 146 |  | add BX, CX |  |
| 147 |  |  |  |
| 148 | ;space |  |  |
| 149 |  | cmp AL, ' ' |  |
| 150 |  | je space |  |
| 151 | ;dot |  |  |
| 152 |  | cmp AL, '.' |  |
| 153 |  | je dot |  |
| 154 | ;lowercase letters |  |  |
| 155 |  | cmp AL, 'a' |  |
| 156 |  | jb t1 |  |
| 157 |  | cmp AL, 'z' |  |
| 158 |  | jbe low |  |
| 159 | ;uppercase letters | ;up | t1: |
| 160 |  | cmp AL, 'A' |  |
| 161 |  | jb save |  |
| 162 |  | cmp AL, 'Z' |  |
| 163 |  | jbe up |  |
| 164 |  |  |  |
| 165 |  |  | space: |
| 166 |  | inc words |  |
| 167 |  | jmp save |  |
| 168 |  | mov idot, 1 | dot: |
| 169 |  | call saveLen |  |
| 170 |  | inc sens |  |
| 171 |  | jmp save |  |
| 172 | ;if last char wasn't dot- save as it is. else- save uppercase | ;if | low: |
| 173 |  | cmp idot, 0 |  |
| 174 |  | je saveL |  |
| 175 |  | sub AL, diff |  |
| 176 |  | jmp saveL |  |
| 177 | ;if last char was dot- save as it is. else- save as it is | ;if | up: |
| 178 |  | cmp idot, 1 |  |
| 179 |  | je saveL |  |
| 180 |  | add AL, diff |  |
| 181 |  | jmp saveL |  |
| 182 |  |  |  |
| 183 |  |  |  |
| 184 |  |  | saveL: |
| 185 |  | mov idot, 0 |  |
| 186 |  |  | save: |
| 187 |  | mov [BX], AL |  |
| 188 |  |  |  |
| 189 | ;check max length |  |  |
| 190 |  | inc count |  |
| 191 |  | cmp count, 200 |  |
| 192 |  | jb char |  |
| 193 |  | ret | en1: |
| 194 |  |  |  |
| 195 |  |  | enter: |
| 196 |  | call saveLen |  |
| 197 |  | ret |  |
| 198 |  |  |  |
| 199 |  |  |  |
| 200 | ; ------------------------------------------------------------ ; | | |
| 201 | outputP: ; output the string(uppercase first letters) ; | | |
| 202 | ; ------------------------------------------------------------ ; | | |
| 203 |  | mov AH, 09h |  |
| 204 |  | lea DX, itxtLn |  |
| 205 |  | int 21h |  |
| 206 |  | ret |  |
| 207 |  |  |  |
| 208 | ; ------------------------------------------------------------ ; | | |
| 209 | saveLen: ; save start point and length at order as array ; | | |
| 210 | ; AL should be return as it is ; | | |
| 211 | ; ------------------------------------------------------------ ; | | |
| 212 |  | mov CL, AL |  |
| 213 |  | mov DX, BX |  |
| 214 |  |  |  |
| 215 |  | inc saves |  |
| 216 |  | lea BX, order |  |
| 217 |  | mov AL, sens |  |
| 218 |  | mov AH, 0 |  |
| 219 |  | add BX, AX |  |
| 220 |  | add BX, AX |  |
| 221 |  | mov AL, point |  |
| 222 |  | mov [BX], AL |  |
| 223 |  | inc BX |  |
| 224 |  | mov AL, len |  |
| 225 |  | mov [BX], AL |  |
| 226 |  | mov AH, point |  |
| 227 |  | add AH, AL |  |
| 228 |  | mov point, AH |  |
| 229 |  | mov len, 0 |  |
| 230 |  |  |  |
| 231 |  | mov BX, DX |  |
| 232 |  | mov AL, CL |  |
| 233 |  | ret |  |
| 234 | ; ------------------------------------------------------------ ; | | |
| 235 | sort: ; sort the lengths array ; | | |
| 236 | ; ------------------------------------------------------------ ; | | |
| 237 |  | mov BX, 0 |  |
| 238 |  | mov CX, 0 |  |
| 239 |  |  | Rche: |
| 240 |  | inc CH |  |
| 241 |  | cmp CH, saves |  |
| 242 |  | je endD |  |
| 243 |  |  | round: |
| 244 |  | lea BX, order |  |
| 245 |  | inc BX |  |
| 246 |  | mov CL, 0 |  |
| 247 |  |  | comp: |
| 248 |  | mov AL, [BX] |  |
| 249 |  | add BX, 2 |  |
| 250 |  | mov AH, [BX] |  |
| 251 | ;if longer or equal- continue, else- replace beween them(point, lenght) |  |  |
| 252 |  | cmp AL, AH |  |
| 253 |  | jae con |  |
| 254 |  | mov [BX], AL |  |
| 255 |  | sub BX, 2 |  |
| 256 |  | mov [BX], AH |  |
| 257 |  |  |  |
| 258 |  | sub BX, 1 |  |
| 259 |  | mov DL, [BX] |  |
| 260 |  | add BX, 2 |  |
| 261 |  | mov DH, [BX] |  |
| 262 |  | mov [BX], DL |  |
| 263 |  | sub BX, 2 |  |
| 264 |  | mov [BX], DH |  |
| 265 |  | add BX, 3 |  |
| 266 |  |  | con: |
| 267 |  | inc CL |  |
| 268 |  | cmp CL, saves |  |
| 269 |  | je Rche |  |
| 270 |  |  |  |
| 271 |  | jmp comp |  |
| 272 |  |  | endD: |
| 273 |  | ret |  |
| 274 | ; ------------------------------------------------------------ ; | | |
| 275 | outputOrder: ; output the string sorted by length ; | | |
| 276 | ; ------------------------------------------------------------ ; | | |
| 277 |  | mov AX, 0 |  |
| 278 |  | mov BX, 0 |  |
| 279 |  | mov CX, 0 |  |
| 280 |  | mov DX, 0 |  |
| 281 |  | mov counter, 0 |  |
| 282 |  |  |  |
| 283 | ;output sentence: | ;out | sen: |
| 284 | ;get starting point |  |  |
| 285 |  | lea BX, order |  |
| 286 |  | add BX, counter |  |
| 287 |  | add BX, counter |  |
| 288 |  | inc counter |  |
| 289 |  | mov AL, [BX] |  |
| 290 |  | mov AH, 0 |  |
| 291 | ;get length |  |  |
| 292 |  | inc BX |  |
| 293 |  | mov CL, [BX] |  |
| 294 |  | mov CH, 0 |  |
| 295 |  |  |  |
| 296 | ;if length is 0- go out |  |  |
| 297 |  | cmp CL, 0 |  |
| 298 |  | je retu1 |  |
| 299 |  |  |  |
| 300 |  | lea BX, itxt |  |
| 301 |  | add BX, AX |  |
| 302 | ;output from starting point to length | ;out | cha: |
| 303 |  | mov AH, 02h |  |
| 304 |  | mov DL, [BX] |  |
| 305 |  | int 21h |  |
| 306 |  |  |  |
| 307 |  | inc BX |  |
| 308 |  | dec CL |  |
| 309 |  | cmp CL, 0 |  |
| 310 |  | je sen |  |
| 311 |  |  |  |
| 312 |  | jmp cha |  |
| 313 |  |  | retu1: |
| 314 |  |  |  |
| 315 |  | ret |  |
| 316 |  |  |  |
| 317 | ; ------------------------------------------------------------ ; | | |
| 318 | outputS: ; binary->ascii, output sentences count ; | | |
| 319 | ; ------------------------------------------------------------ ; | | |
| 320 |  | inc sens |  |
| 321 |  | mov AX, 0 |  |
| 322 |  | mov AL, sens |  |
| 323 |  | dec AL |  |
| 324 |  | mov BL, 100 |  |
| 325 |  | div BL |  |
| 326 |  | add AL, '0' |  |
| 327 |  | mov sen100, AL |  |
| 328 |  | mov AL, AH |  |
| 329 |  | mov AH, 0 |  |
| 330 |  | mov BL, 10 |  |
| 331 |  | div BL |  |
| 332 |  | add AL, '0' |  |
| 333 |  | add AH, '0' |  |
| 334 |  | mov sen10, AL |  |
| 335 |  | mov sen1, AH |  |
| 336 |  |  |  |
| 337 |  | mov AH, 09h |  |
| 338 |  | lea DX, senMsg |  |
| 339 |  | int 21h |  |
| 340 |  | ret |  |
| 341 |  |  |  |
| 342 | ; ------------------------------------------------------------ ; | | |
| 343 | outputW: ; binary->ascii, output words count ; | | |
| 344 | ; ------------------------------------------------------------ ; | | |
| 345 |  | inc words |  |
| 346 |  | mov AX, 0 |  |
| 347 |  | mov AL, words |  |
| 348 |  | mov BL, 100 |  |
| 349 |  | div BL |  |
| 350 |  | add AL, '0' |  |
| 351 |  | mov words100, AL |  |
| 352 |  | mov AL, AH |  |
| 353 |  | mov AH, 0 |  |
| 354 |  | mov BL, 10 |  |
| 355 |  | div BL |  |
| 356 |  | add AL, '0' |  |
| 357 |  | add AH, '0' |  |
| 358 |  | mov words10, AL |  |
| 359 |  | mov words1, AH |  |
| 360 |  |  |  |
| 361 |  | mov AH, 09h |  |
| 362 |  | lea DX, wordsMsg |  |
| 363 |  | int 21h |  |
| 364 |  | ret |  |
| 365 |  |  |  |
| 366 | ; ------------------------------------------------------------ ; | | |
| 367 | repeat: ;ask repeat | | |
| 368 | ; ------------------------------------------------------------ ; | | |
| 369 |  | mov AH, 09h |  |
| 370 |  | lea DX, repeatMsg |  |
| 371 |  | int 21h |  |
| 372 |  |  |  |
| 373 |  | mov AH, 01h |  |
| 374 |  | int 21h |  |
| 375 |  |  |  |
| 376 |  |  |  |
| 377 |  | cmp AL, 'y' |  |
| 378 |  | je main |  |
| 379 |  |  |  |
| 380 |  | cmp AL, 'n' |  |
| 381 |  | je exit |  |
| 382 |  |  |  |
| 383 |  | jmp repeat |  |
| 384 |  |  |  |
| 385 |  |  |  |
| 386 |  |  |  |
| 387 | ;exitMsg, key than exit: |  | ;exitMsg, key than exit: |
| 388 |  |  | exit: |
| 389 |  | mov AH, 09h |  |
| 390 |  | lea DX, exitMsg |  |
| 391 |  | int 21h |  |
| 392 |  |  |  |
| 393 |  | mov AH, 01h |  |
| 394 |  | int 21h |  |
| 395 |  | mov AH, 4Ch |  |
| 396 |  | int 21h |  |
| 397 |  |  | END |