

```

1
2
3 data segment
4
5
6 ;sectiune mesaje
7 randNou db 10,13,32,32,'$'
8 mesaj1 db 'Program de inmultire a 2 matrice cu dimensiunea maxima 10x10 $'
9 mesaj2 db 10,13,'Introduceti numarul de linii al matricei 1: $'
10 mesaj3 db 10,13,'Introduceti numarul de coloane al matricei 1: $'
11 mesaj4 db 10,13,'Introduceti numarul de linii al matricei 2: $'
12 mesaj5 db 10,13,'Introduceti numarul de coloane al matricei 2: $'
13 mesaj6 db 10,13,'Introduceti elementele matricei 1 linie cu linie: $'
14 mesaj7 db 10,13,'Introduceti elementele matricei 2 linie cu linie: $'
15 mesaj8 db 10,13,'Matricea 1 este: $'
16 mesaj9 db 10,13,'Matricea 2 este: $'
17 mesaj_final db 10,13,'Matricea finala este: $'
18 iesi_din_program db 10,10,13,'Eroare. Program terminat xD$'
19
20
21 ;sectiune date matrice
22 mat1 db 100 dup(1)
23 mat2 db 100 dup(1)
24 prod dw 100 dup(0)
25 string db 10 dup('$')
26 linii1 dw ? ; m
27 col1 dw ? ; n
28 linii2 dw ? ; p
29 col2 dw ? ; r
30 dimensiune_max dw 10
31
32 ;sectiune date procedura citeste_numar
33 sir db ?,?
34 msg_citire db 10,13,'Introduceti un numar 0-80: $'
35 ; msg_eroare db 10,13,10,13, 'Eroare. Program terminat xD$'
36 numar db ?
37 contor_eroare db 0
38 maxim db 80
39
40
41 data ends
42
43 cod segment
44
45 ;macro afisare mesaje
46 MESAJ MACRO nume_mesaj ;ce functie jmechera
47     mov dx, offset nume_mesaj
48     mov ah, 9
49     int 21h
50 ENDM
51
52 fara_dimensiune_zero PROC ; procedura de test pentru dimensiuni mat
53     cmp linii1, 0
54     jne bine1
55     EXIT iesi_din_program
56
57     bine1:
58     cmp col1, 0
59     jne bine2
60     EXIT iesi_din_program
61
62     bine2:
63     cmp linii2, 0
64     jne bine3
65     EXIT iesi_din_program
66

```

```

67     bine3:
68     cmp col2,0
69     jne bine4
70     EXIT iesi_din_program
71
72     bine4:
73
74     ENDP fara_dimensiune_zero
75     ret
76
77
78     EXIT MACRO iesi_din_program    ;procedura de eroare
79     mov ah,2
80     mov dx,7                      ;beep
81     int 21h
82     lea dx, iesi_din_program      ;afiseaza eroare
83     mov ah,9
84     int 21h
85     mov ah,4ch                   ;iesi din program
86     int 21h
87     ENDM
88
89     citeste_numar PROC
90
91     xor si,si
92     mov cl,2
93
94     repeta:
95
96     mov ah,1
97     int 21h
98     add contor_eroare,1
99     cmp al,57                    ;testeaza daca am citit cifra, altfel eroare
100    ja eroare
101
102    cmp al,48                      ;testeaza daca am citit cifra
103    jb eroare
104
105    revenire:
106
107    mov sir[si],al                ;stocheaza caracterul ascii citit in sir
108
109    inc si
110    cmp si,2                      ;daca am citit deja 2 cifre, fa conversia
111    je conversie
112    cmp al,13                     ;daca am citit 1 cifra si apas enter, atunci fa conversia
113    jne repeta
114
115    conversie:
116
117    xor ax, ax
118    cmp sir[si-1],13              ;daca am citit doar 1 cifra, sari peste constructie
119    je sari_peste
120    mov al,sir[si-2]              ; daca am citit 2 cifre, atunci construiesc nr
121    sub al, 48
122    mov bl, 10                    ;inmulteste cu 10 pentru a obtine zecile
123    mul bl
124    inc si
125
126    sari_peste:
127    mov bl,sir[si-2]              ;adauga unitatea la nr construit
128    sub bl,48
129    add al, bl
130    mov numar, al                ;nr citit este stocat in "numar"
131
132    mov al,maxim

```

```

133     cmp numar, al      ; daca numarul este mai mare ca limita maxima, eroare
134     ja eroare
135
136     jmp finish
137
138     eroare:            ;citire eronata, iesi din program
139     cmp al,13          ;daca citest o cifra si apas "enter" revino in program
140     je revenire
141     EXIT iesi_din_program
142
143     finish:
144     cmp contor_eroare,2 ;daca am introdus doar 1 enter (fara nicio cifra), eroare
145     je ok
146     cmp al,15          ;conditie veche, nefolosita
147     je eroare
148     cmp al,80
149
150     ok:
151
152     ret
153     ENDP citeste_numar
154
155
156
157
158     ; inceputul efectiv al programului
159     START:
160
161
162     mov ax, data
163     mov ds,ax
164
165     MESAJ mesaj1 ; afisarea functiei programului
166
167     ; citire dimensiuni matrice 1
168     MESAJ mesaj2
169     call citeste_numar ;citeste numar linii matrice1
170     xor ax,ax
171     mov al,numar      ; numar pe 8 biti, liniil pe 16 biti
172     mov liniil, ax
173
174     MESAJ mesaj3
175     call citeste_numar ;citeste numar coloane matrice1
176     xor ax, ax
177     mov al, numar
178     mov col1,ax
179
180     MESAJ mesaj4
181     call citeste_numar ;citeste numar linii matrice2
182     xor ax,ax
183     mov al, numar
184     mov linii2,ax
185
186     MESAJ mesaj5
187     call citeste_numar ;citeste numar coloane matrice2
188     xor ax, ax
189     mov al, numar
190     mov col2,ax
191
192     ;testare conditii dimensiuni matrice
193     call fara_dimensiune_zero ;ma asigur ca dimensiunea matricelor e nenula
194
195     mov ax, linii2      ; nr coloane matrice1 = nr linii matrice2
196     cmp ax,col1
197     je ok1:
198     EXIT iesi_din_program

```

```

199
200 ok1:
201 mov ax, linii1          ; maxim 10 linii si 10 coloane
202 cmp ax,dimensiune_max
203 jle ok2:
204 EXIT iesi_din_program
205
206 ok2:
207 mov ax, col1
208 cmp ax,dimensiune_max
209 jle ok3:
210 EXIT iesi_din_program
211
212 ok3:
213 mov ax, linii2
214 cmp ax,dimensiune_max
215 jle ok4:
216 EXIT iesi_din_program
217
218 ok4:
219 mov ax, col2
220 cmp ax,dimensiune_max
221 jle ok5:
222 EXIT iesi_din_program
223
224 ok5:
225
226 ;introducerea elementelor din matrice
227 MESAJ mesaj6
228 mov ax, linii1          ;calculez nr de elemente din matrice
229 mul col1
230 mov cx,ax
231 xor di, di
232
233
234 bucla_citire1:          ;citesc numerele si le salvez in matrice
235     MESAJ randNou
236     push cx              ;cx este modificat de proc si trebuie stivuit
237     call citeste_numar
238     pop cx
239     mov al, numar
240     mov mat1[di],al      ;salveaza element in matrice
241     inc di
242     mov ah,2
243     mov dx,32            ;veche,nefolosita,afiseaza spatiu intre numere
244     int 21h
245 loop bucla_citire1
246
247 MESAJ mesaj7
248 mov ax,linii2          ;calculez nr de elemente din matrice
249 mul col2
250 mov cx,ax
251 xor di,di
252
253
254 bucla_citire2:
255     MESAJ randNou
256     push cx              ;cx este modificat de proc si trebuie stivuit
257     call citeste_numar
258     pop cx
259     mov al, numar
260     mov mat2[di],al
261     inc di
262     mov ah,2
263     mov dx,32
264     int 21h

```

```

265     loop bucla_citire2
266
267
268
269
270
271     xor si, si ; i, initializari contoare la zero
272     xor di, di ; j
273     xor bx, bx ; k
274     xor cx, cx
275
276     ;ideea calauzitoare:
277     ;prod[i*col2 + j] += mat1[i*col1 + k] * mat2[k*col2 + j];
278
279 bucla_1:
280     xor di, di
281
282
283     bucla_2:
284         xor bx, bx
285
286         bucla_3:
287             push si ;salveaza si ca sa nu stric indexul buclei_1
288             mov ax, si ; pozitioneaza pe linia aferenta
289             mul col1 ; matricei 1
290             mov si, ax
291
292             mov cl, mat1[si][bx]
293             pop si
294
295             push bx
296             mov ax, bx ; pozitioneaza pe linia aferenta
297             mul col2 ; matricei 2
298             mov bx, ax
299
300             mov al, cl
301             mul mat2[bx][di] ;inmultirea elementelor din matrice
302             pop bx
303
304             push si
305             mov cx, ax ; salveza rezultatul inmultirii din ax
306             shl si, 1 ;dubleaza si fiindca mat prod e pe 16 biti
307             mov ax, si
308             mul col2 ;pozitioneaza pe linia corecta din mat prod
309             mov si, ax
310
311             push bx
312             mov bx, di
313             shl bx, 1 ;dubleaza bx fiindca mat prod e pe 16 biti
314             add prod[si][bx], cx ;vezi idee calauzitoare
315             pop bx
316             pop si
317
318
319             inc bx
320             cmp bx, col1
321             jb bucla_3
322
323             inc di
324             cmp di, col2
325             jb bucla_2
326
327             inc si
328             cmp si, liniil
329             jb bucla_1
330

```

```

331 ;afisare matrice1
332 ;functie comentata la afisarea mat prod
333 xor si,si
334 xor di,di
335 mov ax,linii1
336 mul coll
337 mov cx,ax
338
339 MESAJ mesaj8
340 MESAJ randNou
341 repeat_1:
342 push cx
343 xor ax,ax
344 mov al, mat1[si]
345 mov bx, 10
346 mov cx,0
347
348 while1_1:
349 mov dx,0
350 div bx
351 add dx,48
352 push dx
353 inc cx
354 cmp ax,0
355 jne while1_1
356
357 mov bx,offset string
358
359 while2_2:
360 pop dx
361 mov [bx],dx
362 inc bx
363 loop while2_2
364
365 mov ah,9
366 mov dx,offset string
367 int 21h
368
369 inc di
370 mov ax,di
371 xor dx,dx
372 div coll
373 cmp dl,0
374 jne sari_1
375 mov ah,09
376 lea dx,randnou
377 int 21h
378 sari_1:
379 pop cx
380 inc si
381 loop repeat_1
382
383 ;afisare matrice2
384 ;functie comentata la afisarea mat prod
385 xor si,si
386 xor di,di
387 mov ax,linii2
388 mul col2
389 mov cx,ax
390
391 MESAJ mesaj9
392 MESAJ randNou
393 repeat_2:
394 push cx
395 xor ax,ax
396 mov al, mat2[si]

```

```

397     mov bx, 10
398     mov cx, 0
399
400     while1_2:
401     mov dx, 0
402     div bx
403     add dx, 48
404     push dx
405     inc cx
406     cmp ax, 0
407     jne while1_2
408
409     mov bx, offset string
410
411     while2_3:
412     pop dx
413     mov [bx], dx
414     inc bx
415     loop while2_3
416
417     mov ah, 9
418     mov dx, offset string
419     int 21h
420
421     inc di
422     mov ax, di
423     xor dx, dx
424     div col2
425     cmp dl, 0
426     jne sari_2
427     mov ah, 09
428     lea dx, randnou
429     int 21h
430     sari_2:
431     pop cx
432     inc si
433     loop repeat_2
434
435
436     ;afisare matrice finala
437
438     xor si, si
439     xor di, di
440     mov ax, liniil
441     mul col2
442     mov cx, ax
443
444
445     MESAJ mesaj_final
446     MESAJ randNou
447     repeat:
448     push cx                ; salveaza valoarea lui cx
449     mov ax, prod[di]       ; deoarece folosesc 2 bucle imbricate
450
451     mov bx, 10 ;o sa impart nr la 10 pentru a obtine cate o cifra
452
453     mov cx, 0
454
455     while1:
456
457     mov dx, 0
458
459     div bx ;imparte nr la 10 pentru o cifra
460
461     add dx, 48 ;obține codul ascii al cifrei
462

```

```

463 push dx ;salveaza codul ascii pentru afisare
464
465 inc cx
466
467 cmp ax,0 ;cand ajunge nr la zero inseamna ca am obtinut
468 ;toate cifrele
469 jne while1
470
471
472 mov bx,offset string
473
474
475 while2:
476
477 pop dx ;pune in dx codul ascii coresp celei mai
478 ;semnificative cifre
479 mov [bx],dx ;pune in string cifrele una dupa alta
480
481 inc bx
482
483 loop while2
484
485
486 mov ah,09
487
488 mov dx,offset string ;afiseaza cifrele numarului
489
490 int 21h
491
492 add di,2 ;incrementeaza cu 2 fiindca matricea finala
493 inc si ;contine nr pe 16 biti
494
495 mov ax,si
496 xor dx,dx
497 div col2
498 cmp dl,0 ;verifica daca am ajuns la capat de linie
499 jne sari
500 mov ah,09
501 lea dx,randnou ;rand nou daca am terminat linia
502 int 21h
503 sari:
504
505 pop cx ;cx revine la valoarea coresp buclei exterioare
506
507 loop repeat
508
509 mov ah,4ch
510 int 21h
511
512 cod ends
513 end start
514
515
516
517
518
519

```