2d array by using random function

include irvine32.inc

.data

row byte 4

col byte 3

array1 byte 12 DUP(?)

.code

main proc

mov ecx,12

call randomrange

call writedec

call crlf

xor eax,eax

mov al,byte ptr array1

call dumpregs

exit

main endp

end main

reversing array

include irvine32.inc

.data

Array DWORD 1,2,3,4,5,6,7,8,9,10,11,12,13

.code

main PROC

mov esi, OFFSET Array

mov edi, OFFSET Array

add edi, SIZEOF Array

sub edi, TYPE Array

mov ecx, LENGTHOF Array

L2:

mov eax, [edi]

mov eax, [edi]

xchg ebx, ebx

mov [esi], ebx

mov [edi], ebx

add esi, TYPE Array

SUB edi, TYPE Array

dec ecx

LOOP L2

xor eax,eax

mov ax,word ptr Array

call dumpregs

main ENDP

END main

Transpose of matrix

include irvine32.inc

.data

Array DWORD 3 dup(3 dup(?))

i byte 0

j byte 0

count byte 0

.code

main PROC

mov ecx,4

outer:

mov ecx,3

inner:

inc i

inc j

Array[i+j+k]

inc count

LOOP inner

mov ecx,count

LOOP outer

call dumpregs

main ENDP

END main

2d array on run time and multiplication of 2 matrix

And resultant of matrix

include irvine32.inc

.data

array1 byte 3 DUP(4 DUP (?))

array2 byte 1,2,3,4,5,6,7,8,9,10,11,12

var1 byte 0

array3 byte 4 DUP(4 DUP(0))

.code

main proc

xor eax,eax

call readint

mov byte ptr array1,al

mov al, byte ptr array1

;xor esi , esi

;mov esi,OFFSET array

call readint

mov byte ptr array1+1,al

mov al, byte ptr array1+1

;mov esi,OFFSET array+1

call readint

mov byte ptr array1+2,al

mov al, byte ptr array1+2

;mov esi,OFFSET array+2

call readint

mov byte ptr array1+3,al

mov al, byte ptr array1+3

;mov esi,OFFSET array+3

call readint

mov byte ptr array1+4,al

mov ah, byte ptr array1+4

;mov esi,OFFSET array+4

call readint

mov byte ptr array1+5,al

mov ah, byte ptr array1+5

;mov esi,OFFSET array+5

call readint

mov byte ptr array1+6,al

mov ah, byte ptr array1+6

;mov esi,OFFSET array+6

call readint

mov byte ptr array1+7,ah

mov ah, byte ptr array1+7

;mov esi,OFFSET array+7

call readint

mov byte ptr array1+8,ah

mov ah, byte ptr array1+8

;mov esi,OFFSET array+8

call readint

mov byte ptr array1+9,ah

mov ah, byte ptr array1+9

;mov esi,OFFSET array+9

call readint

mov byte ptr array1+10,ah

mov ah, byte ptr array1+10

;mov esi,OFFSET array+10

call readint

mov byte ptr array1+11,ah

mov ah, byte ptr array1+11

;mov esi,OFFSET array+11

call readint

mov byte ptr array1+12,ah

mov ah, byte ptr array1+12

;mov esi,OFFSET array+12

mov al,byte ptr array1

mul byte ptr array2

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1+1

mul byte ptr array2 + 4

add var1,ah

mov al, byte ptr array1 + 3

mul byte ptr array2 + 8

add var1,al

mov al,var1

mov byte ptr array3 , al

mov var1,0

mov al,byte ptr array1

mul byte ptr array2 + 1

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1+1

mul byte ptr array2 + 5

add var1,ah

mov al, byte ptr array1 + 3

mul byte ptr array2 + 9

add var1,al

mov al,var1

mov byte ptr array3 + 1 , al

mov var1,0

mov al,byte ptr array1

mul byte ptr array2 + 2

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1+1

mul byte ptr array2 + 6

add var1,ah

mov al, byte ptr array1 + 2

mul byte ptr array2 + 10

add var1,al

mov al,var1

mov byte ptr array3 + 2 , al

mov var1,0

mov al,byte ptr array1

mul byte ptr array2+3

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1+1

mul byte ptr array2 + 7

add var1,ah

mov al, byte ptr array1 + 2

mul byte ptr array2 + 11

add var1,al

mov al,var1

mov byte ptr array3 + 3 , al

mov var1,0

mov al,byte ptr array1 + 3

mul byte ptr array2

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 4

mul byte ptr array2 + 4

add var1,ah

mov al, byte ptr array1 + 5

mul byte ptr array2 + 8

add var1,al

mov al,var1

mov byte ptr array3 + 4 , al

mov var1,0

mov al,byte ptr array1 + 3

mul byte ptr array2 + 1

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 4

mul byte ptr array2 + 5

add var1,ah

mov al, byte ptr array1 + 5

mul byte ptr array2 + 9

add var1,al

mov al,var1

mov byte ptr array3 + 5 , al

mov var1,0

mov al,byte ptr array1 + 3

mul byte ptr array2 + 2

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 4

mul byte ptr array2 + 6

add var1,ah

mov al, byte ptr array1 + 5

mul byte ptr array2 + 10

add var1,al

mov al,var1

mov byte ptr array3 + 6 , al

mov var1,0

mov al,byte ptr array1+ 3

mul byte ptr array2+3

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 4

mul byte ptr array2 + 7

add var1,ah

mov al, byte ptr array1 + 5

mul byte ptr array2 + 11

add var1,al

mov al,var1

mov byte ptr array3 + 7 , al

mov var1,0

mov al,byte ptr array1 + 6

mul byte ptr array2

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 7

mul byte ptr array2 + 4

add var1,ah

mov al, byte ptr array1 + 8

mul byte ptr array2 + 8

add var1,al

mov al,var1

mov byte ptr array3 + 8 , al

mov var1,0

mov al,byte ptr array1 + 6

mul byte ptr array2 + 1

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 7

mul byte ptr array2 + 5

add var1,ah

mov al, byte ptr array1 + 8

mul byte ptr array2 + 9

add var1,al

mov al,var1

mov byte ptr array3 + 9 , al

mov var1,0

mov al,byte ptr array1 + 6

mul byte ptr array2 + 2

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 7

mul byte ptr array2 + 6

add var1,ah

mov al, byte ptr array1 + 8

mul byte ptr array2 + 10

add var1,al

mov al,var1

mov byte ptr array3 + 10 , al

mov var1,0

mov al,byte ptr array1 + 6

mul byte ptr array2+3

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 7

mul byte ptr array2 + 7

add var1,ah

mov al, byte ptr array1 + 8

mul byte ptr array2 + 11

add var1,al

mov al,var1

mov byte ptr array3 + 11, al

mov var1,0

mov al,byte ptr array1 + 9

mul byte ptr array2

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 10

mul byte ptr array2 + 4

add var1,ah

mov al, byte ptr array1 + 11

mul byte ptr array2 + 8

add var1,al

mov al,var1

mov byte ptr array3 + 12 , al

mov var1,0

mov al,byte ptr array1 + 9

mul byte ptr array2 + 1

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 10

mul byte ptr array2 + 5

add var1,ah

mov al, byte ptr array1 + 11

mul byte ptr array2 + 9

add var1,al

mov al,var1

mov byte ptr array3 + 13 , al

mov var1,0

mov al,byte ptr array1 + 9

mul byte ptr array2 + 2

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 10

mul byte ptr array2 + 6

add var1,ah

mov al, byte ptr array1 + 11

mul byte ptr array2 + 10

add var1,al

mov al,var1

mov byte ptr array3 + 14 , al

mov var1,0

mov al,byte ptr array1 + 9

mul byte ptr array2+3

add var1,al

;MOV byte ptr array3,al

mov ah,byte ptr array1 + 10

mul byte ptr array2 + 7

add var1,ah

mov al, byte ptr array1 + 11

mul byte ptr array2 + 11

add var1,al

mov al,var1

mov byte ptr array3 + 15 , al

xor eax,eax

mov al ,byte ptr array3

call dumpregs

exit

main endp

end main