**את"מ מעבדה 10:  
  
מגישים:  
 אורי מלכא- 314862996   
אלן ציפין- 313206062**;

; lab10 ori and alan

; int initarr(int\*\*arr, int n, int (\*initfunc)(int));

.MODEL SMALL

.STACK 100h

.DATA

arrsize DW 0

TWO DB 2

halfsize DW 0

.CODE

\_initarr PROC NEAR

PUBLIC \_initarr

EXTRN \_malloc : NEAR

EXTRN \_rand : NEAR

;save values

PUSH BP

PUSH SI

PUSH DI

MOV BP,SP

; int\*\*arr=[BP+8]

; int n = [BP+10]

; int (\*initfunc)(int)) = [BP+12]

MOV AX, word PTR [BP+10]

MOV SI,[BP+8]

MOV arrsize,AX

ADD AX,AX ; AX=n\*2 each

MOV DI,[BP+12] ;

PUSH BX

PUSH CX

PUSH DX

PUSH AX

CALL \_malloc

POP DX ; We dont care where to push the saved value, so we run over it.

POP DX

POP CX

POP BX

MOV [SI],AX ;// \*arr=malloc(n\*sizeof(int\*))

CMP AX,0

JE MemoFailed

MOV AX,[BP+10]

DIV TWO ;AL=N/2

XOR AH,AH

MOV halfsize,AX

MOV BX,[SI]

;Once we got here, halfsize=N/2 ,BX is the pointer to the allocated array , DI=int (\*initfunc)(int))

XOR CX,CX

GetNumArray:

CMP CX,halfsize ; here we stop when we called the function n/2 times

JE GetNumArrayFinish

PUSH CX

CALL DI

MOV [BX],AX ; DI return int from the function

ADD BX,2 ;arr++

POP CX

INC CX

JMP GetNumArray

GetNumArrayFinish:

; Once we got here CX=N/2 BX is pointing to arr[n/2]

RandArray:

CMP CX,arrsize ; here we stop when we called the function n/2 times

JE RandArrayFinish

; Save BX,DX,CX

PUSH BX

PUSH DX

PUSH CX

CALL \_rand

POP CX

POP DX

POP BX

MOV [BX],AX ; DI return int from the function

ADD BX,2 ;arr++

INC CX

JMP RandArray

RandArrayFinish:

MOV AX,1 ;// if we got here the memo allocation succeeded

MemoFailed: ; if we jmped to this label its when AX=0 (memo failed)

POP DI

POP SI

POP BP

RET

\_initarr ENDP

END