

Task#1:

Take an array of 10 numbers move word-type of data into another empty array using stack push and pop technique.

```
1  TITLE TASK 1
2  INCLUDE Irvine32.inc
3  .data
4  arr WORD 10h,20h,30h,40h,50h,60h,70h,80h,90h,1Ah
5  arr2 WORD 10 dup(?)
6
7  .code
8  main PROC
9  mov eax, 0
10 mov esi, OFFSET arr
11 mov edi, OFFSET arr2
12 mov ecx, LENGTHOF arr
13 L1:
14 mov al, [esi]
15 push eax
16 add esi, TYPE arr
17 loop L1
18 mov ecx, LENGTHOF arr
19 L2:
20 pop eax
21 mov [edi], al
22 add edi, TYPE arr2
23 loop L2
24 mov esi, OFFSET arr2
25 mov ecx, LENGTHOF arr2
26 L3:
27 mov al, [esi]
28 call WriteHex
29 add esi, TYPE arr2
30 call Crlf
31 loop L3
32 exit
33 main ENDP
34 END main
```

```
0000001A
00000090
00000080
00000070
00000060
00000050
00000040
00000030
00000020
00000010

C:\Users\k230842\source
To automatically close
```

Task#2

Write a program which displays the addition of three integers through a stack.

```
1  TITLE TASK 2
2  INCLUDE Irvine32.inc
3  .data
4  var1 DWORD 3h
5  var2 DWORD 2h
6  var3 DWORD 1h
7  .code
8  main PROC
9  push var1
10 push var2
11 push var3
12 call AddSum
13 call WriteHex
14 call DumpRegs
15 exit
16 main ENDP
17 AddSum PROC
18 mov eax, 0
19 push ebp
20 mov ebp, esp
21 add eax, [ebp+8]
22 add eax, [ebp+12]
23 add eax, [ebp+16]
24 pop ebp
25 ret 12
26 AddSum ENDP
27 END main
```

3+2+1 = 6 hence EAX will show 00000006.

```
00000006
EAX=00000006 EBX=003EA000 ECX=004610AA EDX=004610AA
ESI=004610AA EDI=004610AA EBP=0056FA50 ESP=0056FA44
EIP=00463681 EFL=00000202 CF=0 SF=0 ZF=0 OF=0 AF=0 PF=0

C:\Users\k230842\source\repos\Coallab7\Debug\Coallab7.exe (proces
```

Task#3

Write a program having nested procedures are used to calculate the total sum of 2 arrays (each array having 5-elements). The sum of 1-array in 1st procedure and in 2nd procedure have sum of 2-array. And the 3rd procedure added the results of both.

```
TITLE TASK 3
INCLUDE Irvine32.inc
.data
    arr1 word 1,2,3,4,5
    arr2 word 6,7,8,9,10
    sum1 word 0
    sum2 word 0
    totalSum word 0
    total BYTE "The total sum of both arrays (arr1+arr2) is: ",0
.code
; Nested Procedures main ->sumArr1 -> sumArr2 -> addSum ->ret

main PROC
    call sumArr1
    mov edx, OFFSET total
    call WriteString
    mov ax, totalSum
    call WriteDec
    call Crlf
    exit
main ENDP
```

```
sumArr1 PROC
    mov esi, OFFSET arr1
    mov ecx, LENGTHOF arr1
    mov eax,0
l1:
    add ax, [esi]
    add esi, 2
    loop l1
    mov sum1, ax

    call sumArr2

    ret
sumArr1 ENDP
```

```

sumArr2 PROC
    mov esi, OFFSET arr2
    mov ecx, LENGTHOF arr2
    mov eax, 0
l1:
    add ax, [esi]
    add esi, 2
    loop l1
    mov sum2, ax

    call AddSum

    ret
sumArr2 ENDP

```

```

AddSum PROC
    mov ax, sum1
    add ax, sum2
    mov totalSum, ax
    ret
AddSum ENDP

END main

```

```

The total sum of both arrays (arr1+arr2) is: 55
C:\Users\k230842\source\repos\Coallab7\Debug\Coallab7.exe
To automatically close the console when debugging st

```

Task#4

Print the following pattern using a function call in which number of columns is pass through a variable.

```
*  
**  
***  
****  
*****
```

```
TITLE TASK 4  
INCLUDE Irvine32.inc  
.data  
numcolumns DWORD ?  
.code  
main PROC  
mov [numcolumns], 5  
call PrintPattern  
exit  
main ENDP
```

```

PrintPattern PROC
mov ecx, 1
L1:
push ecx
mov edx, [numcolumns]
sub edx, ecx
    printspaces:
    cmp edx, 0
    je printstars
    mov al, ' '
    call WriteChar
    dec edx
    jmp printspaces
    printstars:
    mov ebx, ecx
    printstarloop:
    mov al, '*'
    call WriteChar
    dec ebx
    jnz printstarloop
    call Crlf
pop ecx
inc ecx
cmp ecx, [numcolumns]
jle L1
ret
PrintPattern ENDP

```

```

END main

```

```

*
**
***
****
*****

```

C:\Users\k230842\source\repo

Task#5

Print the following pattern using a function call in which number of columns is pass through a variable.

A
BC
DEF
GHIJ
KLMN

```
TITLE PRINT PATTERN
INCLUDE Irvine32.inc
.data
numcolumns DWORD ?
index DWORD 0
arr BYTE "ABCDEFGHIJKLMNOPQRSTUVWXYZ",0
.code
main PROC
mov [numcolumns], 5
push numcolumns
call PrintPattern
exit
main ENDP
```

```

PrintPattern PROC
mov esi, OFFSET arr
push ebp
mov ebp, esp
mov ecx, 1
mov al, 'A'
L1:
push ecx
mov edx, [ebp+8]
sub edx, ecx
    printspaces:
    cmp edx, 0
    je printstars
    mov al, ' '
    call WriteChar
    dec edx
    jmp printspaces
    printstars:
    mov ebx, ecx
    printstarloop:
    mov al, [esi]
    call WriteChar
    inc esi
    dec ebx
    jnz printstarloop

```

```

call Crlf
pop ecx
inc ecx
cmp ecx, [ebp+8]
jle L1
pop ebp
ret

```

```

PrintPattern ENDP

```

```

END main

```



```
t
  A
  BC
  DEF
  GHIJ
  KLMNO
C:\Users\k230842\sou
```

Task#6

Write a function that asks the user for a number n and prints the sum of the numbers 1 to n

```
TITLE TASK 6
INCLUDE Irvine32.inc
.data
n DWORD ?
sum DWORD 0
msg BYTE "Enter value of n: ",0
total BYTE "The sum of 1 to n numbers is: ",0
.code
main PROC
mov eax, 0
mov edx, OFFSET msg
call WriteString
call Crlf
call Readint
mov [n], eax
call Sum1TON
mov edx, OFFSET total
call WriteString
mov [sum], eax
call WriteDec
call DumpRegs
exit
main ENDP
```

```
Sum1TON PROC
mov eax, 0
mov ecx, [n]
L1:
add eax, ecx
dec ecx
jnz L1
mov [sum], eax
ret
Sum1TON ENDP
END main
```

```
Enter value of n:
5
The sum of 1 to n numbers is: 15
EAX=0000000F  EBX=005A0000  ECX=00000000  EDX=00B3601B
ESI=00B310AA  EDI=00B310AA  EBP=006FF868  ESP=006FF85C
EIP=00B3369C  EFL=00000202  CF=0   SF=0   ZF=0   OF=0   AF=0   PF=0

C:\Users\k230842\source\repos\Coallab7\Debug\Coallab7.exe (process
```

5+4+3+2+1 = 15.