TASK 1 FIBIONACCI SEQUENCE.

1, 1, 2, 3, 5, 8, 13, ...,

```
mov eax, 1
mov ebx, 1
MOV ecx; 8
fibloop:
    mov edx, eax
    add eax, ebx
    mov ebx, edx
    call DumpRegs
    dec ecx
    jnz fibloop
```

eax = 2, 3

eax = 2

ecx = 7

jumps out of loop if ecx = = 0.

23K-0842

TASK 4 COPYING A STRING

source Byte "This is me Kinza", 0

target Byte SIZEOF source DUP(0)

length * type

all characters of * 1

string source

logic :- (1) mov ecx, SIZEOF source

(2) Iterate each character of source string and then copy that character into target string.

(3) Increment esi so that it moves to position of next character.

(4) To print target string

mov edx, OFFSET target

call WriteString

TASK 5 23k-0842

Reverse an Array

DWORD 0 4 8 12 16 20 24 28

1, 2, 3, 4, 5, 6, 7, 8

esi → mov esi, OFFSET arr

edi → mov edi, OFFSET arr

add edi, arrsize - elementsize

arrsize = SIZEOF arr = length of * type

= 8 * 4

= 32

elementsize = TYPE arr

DWORD → 4 bytes

32 - 4 = 28

edi points to the last element.

reverseloop:

cmp esi, edi

1, 2, 3, 4, 5, 6, 7, 8

esi edi

edi esi

jge done

If esi is greater than or equal to edi it means that esi and edi have crossed each other. So it will jump to done label.

code for swapping

```
mov eax, [esi]
mov ebx, [edi]
mov [esi], ebx
mov [edi], eax
```

```
add esi, elementsize
sub edi, elementsize
jmp reverseloop          [unconditional]
```

loop will only exit if esi >= edi

```
done :
        mov esi, OFFSET arr
        mov ecx, arrcount
        printloop :
            mov eax, [esi]
            call WriteDec
            call Crlf
            add esi, elementsize
        loop printloop
exit
main ENDP
END main
```

8 7 6 5 4 3 2 1

printing array

8 → call Crlf → 7 → 6 → 5 → 4 → 3 → 2 → 1

different line

[esi] → 1

Initially esi → address of first element of arr.

edi → address of last element of arr.

[edi] → 8

```
mov eax, [esi]
mov ebx, [edi]
mov [esi], ebx
mov [edi], eax
add esi, elementsize
sub edi, elementsize
```

1 8 2 7 3 6 4 5 5 4 6 3 7 2 8 1

esi ↑ edi ↑

Break condition

eax = 1 → [esi] = ebx

ebx = 8 → [edi] = 1

Break condition :-

```
cmp esi, edi
jge done
```

esi → edi

6 5

edi esi x

done.

23K-0842

TASK 6

To increase efficiency

arr WORD 2, 4, 6, 5

2, 4, 5, 6

only one swap needed.

1st time mai hi sorting hogaegi.

Ek dafa chalaya loop aur agar koi change nhi aya toe iska matlab sorting already hochuke hai.

Toe we need to change the conditions.

```
arr   WORD  6 , 2, 4, 5
swap  BYTE  0    ; use this as a flag

outerloop
   mov  bx, 0
   mov  [swap], 0
   or
   mov  BYTE  [swap] , 0
   innerloop:
   mov  ax , [arr + bx]
   cmp  ax , [arr + bx +2]
   jbe  noswap
        ;swap
         mov  dx, [arr +bx + 2]
         mov  [arr +bx+2], ax
         mov  [arr +bx ], dx
         mov  Byte [swap], 1
```

```
noswap:
    mov add bx, 2
        cmp bx, 6
        jne innerloop
;compare k swap hua hai k nhi.
 cmp   BYTE : [swap], 1
 je   outerloop
```

TASK 2:- PATTERN PRINTING

```
numlines    DWORD   4

mov     ecx , numlines      ; 4
mov     ebx , numlines      ; 4

outerloop :
    mov     edx , ebx       ; 4
    mov     esi , 1         ; 1

    innerloop :
        mov     eax , esi   ; 1
        call    WriteDec    ; eax = 1
        inc     esi         ; esi = 2
        dec     edx
    jnz     innerloop       ; jumps back to inner loop if edx is not zero

    call    Crlf            ; If edx == 0
    dec     ebx
loop    outerloop
```

```
1 2 3 4
1 2 3
1 2
1
```