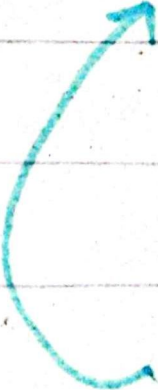


## Task 1 FIBONACCI SEQUENCE.

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

```
mov eax, 1
mov ebx, 1
mov ecx, 8
fibloop:
    mov edx, eax
    add eax, ebx    = 2    eax=2, 3
    mov ebx, edx
    call DumpRegs   eax=2
    dec ecx          ecx=7
    jnz fibloop
```



jumps out of loop if  $ecx == 0$ .

23K-0842

## TASK 4      COPYING A STRING

source Byte      "this is me künza", 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 write string



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

$$\begin{aligned} \text{arrsize} &= \text{sizeof arr} = \text{length of type} \\ &= 8 * 4 \\ &= 32 \end{aligned}$$

elementsize = TYPE arr

DWORD → 4 bytes

$$32 - 4 = \boxed{28}$$

edi points to the last<sup>^</sup> element

reverse loop:

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.

mov eax, [esi]

mov ~~edi~~ ebx, [edi]

mov [esi], ebx

mov [edi], eax

code for  
swapping



```
add esi, elementsize
sub edi, elementsize
```

```
jmp reverseloop. [unconditional]
```

loop will only exit if  $esi \geq edi$

done :

```
mov esi, OFFSET arr
```

```
mov ecx, arrcount
```

printloop :

```
mov eax, [esi]
```

```
call WriteDec
```

```
call Crlf
```

```
add esi, elementsize
```

```
loop printloop
```

printing array

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

call Crlf  
different line

exit

main ENDP

END main

Initially  $esi \rightarrow$  address of first element of arr.  
 $edi \rightarrow$  address of last element of arr.

```
mov eax, [esi]
```

```
mov ebx, [edi]
```

```
mov [esi], ebx
```

```
mov [edi], eax
```

```
add esi, elementsize
```

```
sub edi, elementsize
```

Break condition

$esi = 1$   $ebx = 8$   
 $[esi] = ebx$   $[edi] = 1$

Break condition :-

```
cmp esi, edi
```

```
jge done
```

esi  $\rightarrow$  edi

edi  $\rightarrow$  esi  
done.



TASK 6

To increase efficiency

arr WORD 2, 4, 6, 5  
2, 4, 5, 6

only one swap needed.

1st time mai hi sorting hogaegei.

Ek dafa chalaya loop aur agar koi change nhi aya  
toe iska matlab sorting already ho chuke 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

mov ebx, numlines

outerloop:

mov edx, ebx

mov esi, 1

innerloop:

mov eax, esi

call WriteDec       $eax = 1$

inc esi       $esi = 2$

dec edx

If  $edx == 0$        $jnz$  innerloop      → jumps back to inner loop if edx is not zero

call crlf

dec ebx

loop outerloop

1 2 3 4

1 2 3

1 2

1