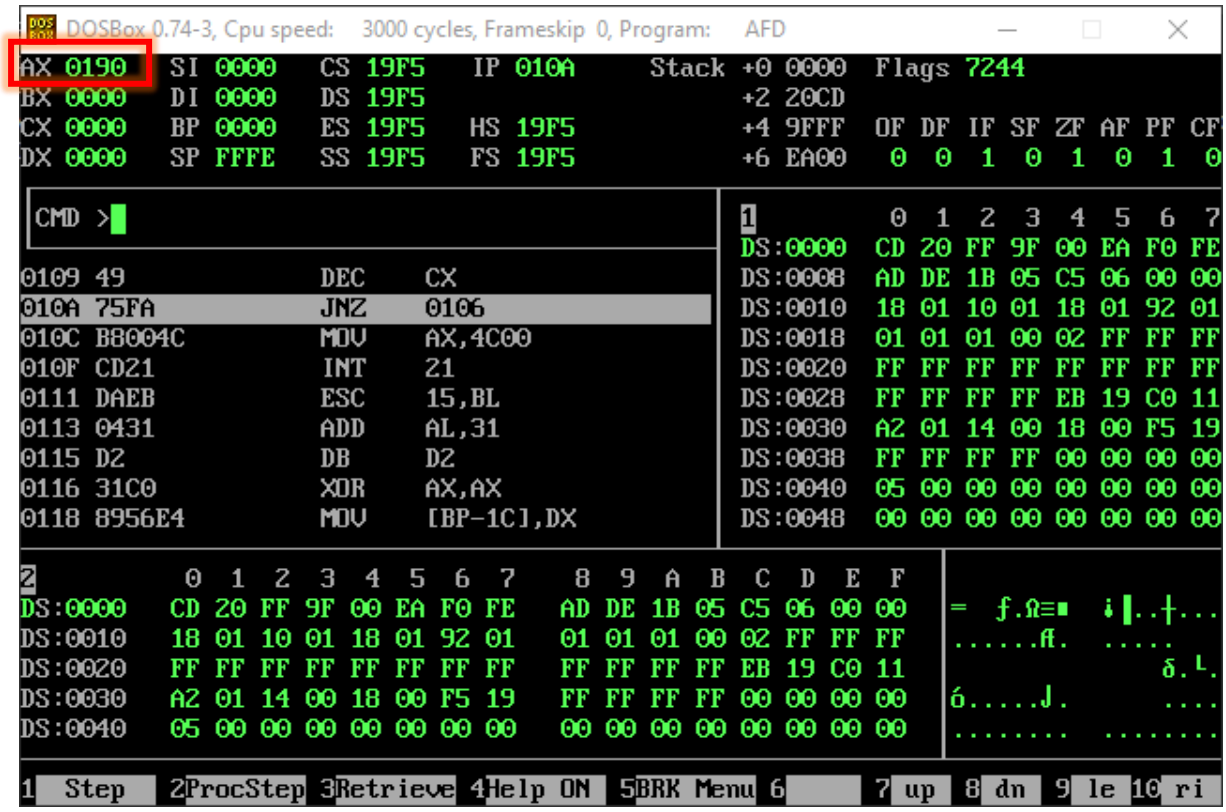
Question No. 1

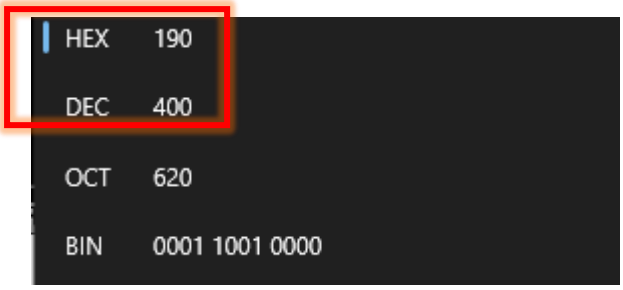
➢ Code Screen Shot

```
[org 0x0100]

mov ax , 0
mov cx , 20

loop1:  add ax , 20
        dec cx

        jnz loop1

mov ax , 0x4c00
int 0x21
```

➢ Final Answer In Debugger



➢ Checking The Answer with Calculator



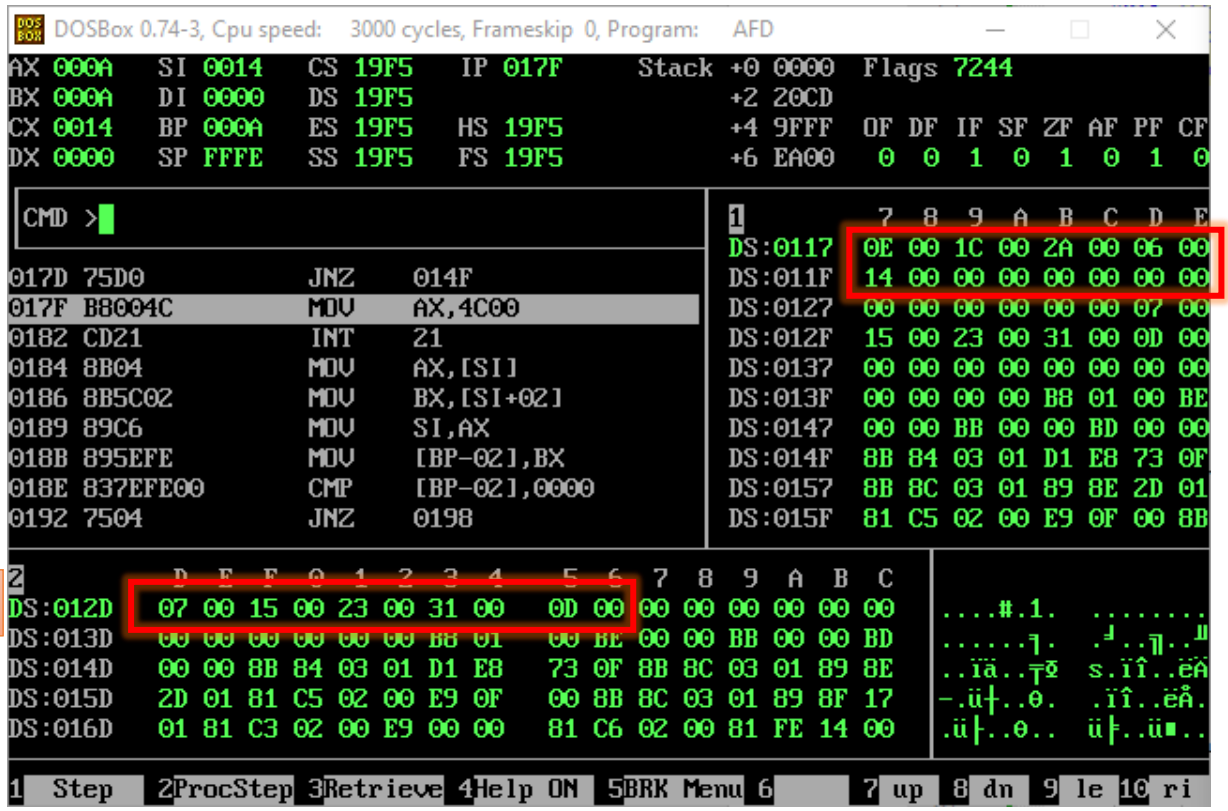
Which Is the Square Of 20 And Got It by Adding 20 20 times in AX

Question No. 2

➢ Code Screen Shot

```
[org 0x0100]

jmp start

array : dw 7 , 14 , 21 , 28 , 35 , 42 , 49 , 6 , 13 , 20
even1 : dw 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0
odd1 : dw 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0

start:
    mov ax , 1
    mov si , 0
    mov bx , 0
    mov bp , 0

    loop1:  mov ax , [array + si]
            shr ax , 1

            jnc evenInstructions

            mov cx , [array + si]
            mov [odd1 + bp] , cx
            add bp , 2
            jmp endOfLoop

        evenInstructions:

            mov cx , [array + si]
            mov [even1 + bx] , cx
            add bx , 2

            jmp endOfLoop

        endOfLoop
            add si , 2
            cmp si , 20
    jnz loop1

mov ax , 0x4c00
int 0x21
```

➢ Final Answers in Debugger



➢ Explanation

I've done Question 2 in a manner that I'm accessing each number in the array and then performing right shift that causes the most right bit of that number to fall into the carry. And as we know the right most bit decides whether the number is odd or even. If the carry due to the right shift is 1 then it means that the number is odd or if the carry is 0 then that means that it is even and then I'm finally storing them into their respective arrays as named in the code. This process happens until all the numbers in the given array are checked and distributed into the even and odd arrays.