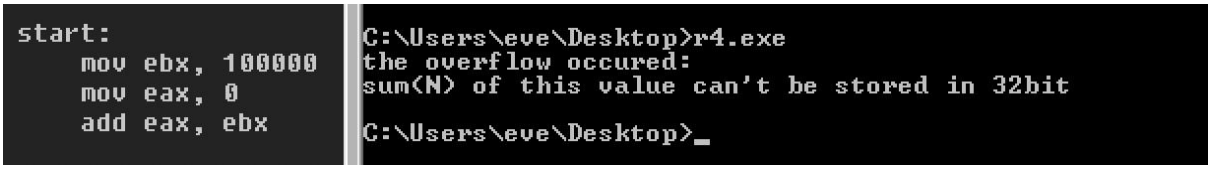
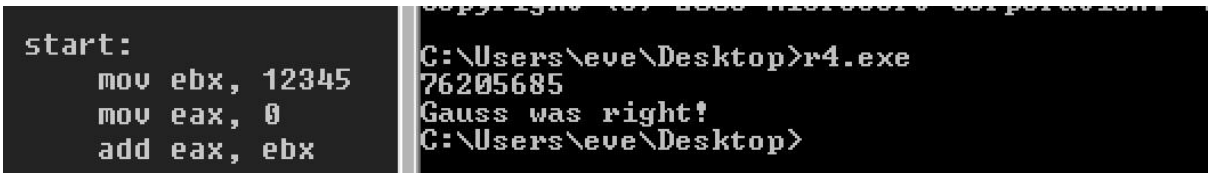
## Task 1 rewritten

1. Screenshots for Task 1.









```
start:
    mov ebx, 0
    mov eax, 0
```

```
C:\Users\eve\Desktop>r4.exe
N must be greater than 0
```

```
start:
    mov ebx, -1
    mov eax, 0
    add eax, ebx
```

```
C:\Users\eve\Desktop>
C:\Users\eve\Desktop>r4.exe
N must be greater than 0

C:\Users\eve\Desktop>
```

I also added comments on this task:

```
start:
    mov ebx, 100        ; N = 100
    mov eax, 0
    add eax, ebx        ; i = N

    mov ecx, 0
    add ecx, ebx        ; j = N

    mov edx, 0
    add edx, ebx        ; k = N

    cmp ebx, 0          ; if N <= 0: call warning, this won't have any reasonable fot this task result
    jle warning0

sumN1:
    dec ebx             ; N -= 1
    add eax, ebx        ; i += N
    cmp eax, 2147483647 ; if N > 65536, then sum(N) > 2147483647, it will cause overflow for signed registers
    jo warning          ; so we compare i to 2147483647 to prevent overflow
    cmp ebx, 0          ; if N != 0, loop must be processed again
    jne sumN1           ; when N = 0, it means that i got sum(N)
    mov ebx, eax        ; N = i; we need it because i (EAX), will be rewritten in further code

 sumN2:
    inc ecx             ; j += 1
    mov eax, ecx        ; i = j
    mul edx             ; i *= k

    shr eax, 1          ; i /= 2, or shift EAX right by 1 bit
    shl edx, 31         ; shift edx left by 31 bit
    add eax, edx        ; ^^^ this is basically division on 64-bit integer by 2

    jo warning          ; call warning if overflow happens
    cmp eax, ebx        ; if sumN1 = sumN2, task completed successfully
    je end_sum
    jne end_wrong

end_wrong:
    print str$(edx), 13, 10, 0
    print "What happened?.. Was Gauss wrong?.."
    ret

end_sum:
    print str$(eax), 13, 10, 0
    print "Gauss was right!"
    ret

warning:
    print "the overflow occured: ", 13, 10, 0
    print "sum(N) of this value can't be stored in 32bit", 13, 10, 0
    ret

warning0:
    print "N must be greater than 0", 13, 10, 0
    ret
END start
```