Assignment 1

1. Screenshots for Task 1.

```
start:
    mov ebx, 100000
    mov eax, 0
    add eax, ebx
```
```
C:\Users\eve\Desktop>r4.exe
the overflow occured:
sum(N) of this value can't be stored in 32bit

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

```
start:
    mov ebx, 65536
    mov eax, 0
    add eax, ebx
```
```
Copyright (c) 2006 Microsoft Corporation.  All right

C:\Users\eve\Desktop>r4.exe
the overflow occured:
sum(N) of this value can't be stored in 32bit
```

```
start:
    mov ebx, 65535
    mov eax, 0
    add eax, ebx
```
```
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.

C:\Users\eve\Desktop>r4.exe
2147450880
Gauss was right!
C:\Users\eve\Desktop>
```

```
start:
    mov ebx, 12345
    mov eax, 0
    add eax, ebx
```
```
Copyright (c) 2006 Microsoft Corporation.

C:\Users\eve\Desktop>r4.exe
76205685
Gauss was right!
C:\Users\eve\Desktop>
```

```
start:
    mov ebx, 999
    mov eax, 0
    add eax, ebx
```
```
Copyright (c) 2006 Microsoft C

C:\Users\eve\Desktop>r4.exe
499500
Gauss was right!
C:\Users\eve\Desktop>
```

```
start:
    mov ebx, 100
    mov eax, 0
    add eax, ebx
```
```
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.

C:\Users\eve\Desktop>r4.exe
5050
Gauss was right!
```

```
start:
    mov ebx, 42
    mov eax, 0
    add eax, ebx
```
```
C:\Users\eve\Desktop>r4.exe
903
Gauss was right!
C:\Users\eve\Desktop>
```

```
start:
    mov ebx, 5
    mov eax, 0
    add eax, ebx
```
```
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.

C:\Users\eve\Desktop>r4.exe
15
Gauss was right!
C:\Users\eve\Desktop>
```

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

```
Copyright (c) 2008 Microsoft Corporatio
C:\Users\eve\Desktop>r4.exe
N must be greater than 0
```

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

```
Copyright (c) 2008 Microsoft Corporatio
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:

```
13    start:
14        mov ebx, 100        ; N = 100
15        mov eax, 0
16        add eax, ebx        ; i = N
17
18        mov ecx, 0
19        add ecx, ebx        ; j = N
20
21        mov edx, 0
22        add edx, ebx        ; k = N
23
24        cmp ebx, 0          ; if N <= 0: call warning, this won't have any reasonable fot this task result
25        jle warning0
26
27    sumN1:
28        dec ebx             ; N -= 1
29        add eax, ebx        ; i += N
30        cmp eax, 2147483647 ; if N > 65536, then sum(N) > 2147483647, it will cause overflow for signed registers
31        jo warning          ; so we compare i to 2147483647 to prevent overflow
32        cmp ebx, 0          ; if N != 0, loop must be processed again
33        jne sumN1           ; when N = 0, it means that i got sum(N)
34        mov ebx, eax        ; N = i; we need it because i (EAX), will be rewritten in further code
35
36    sumN2:
37        inc ecx             ; j += 1
38        mov eax, ecx        ; i = j
39        mul edx             ; i *= k
40
41        shr eax, 1          ; i /= 2, or shift EAX right by 1 bit
42        shl edx, 31         ; shift edx left by 31 bit
43        add eax, edx        ; ^^^ this is basically division on 64-bit integer by 2
44
45        jo warning          ; call warning if overflow happens
46        cmp eax, ebx        ; if sumN1 = sumN2, task completed successfully
47        je end_sum
48        jne end_wrong
49
50    end_wrong:
51        print str$(edx), 13, 10, 0
52        print "What happened?.. Was Gauss wrong?.."
53        ret
54
55    end_sum:
56        print str$(eax), 13, 10, 0
57        print "Gauss was right!"
58        ret
59
60    warning:
61        print "the overflow occured: ", 13, 10, 0
62        print "sum(N) of this value can't be stored in 32bit", 13, 10, 0
63        ret
64
65    warning0:
66        print "N must be greater than 0", 13, 10, 0
67        ret
68    END start
```

2. Screenshots for Task 2.

```
strcreator PROTO :DWORD, :DW

.data
    n dd 4294967295
.data?
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.0
Copyright (c) 2006 Microsoft (

C:\Users\eve\Desktop> t2.exe
4294967295
C:\Users\eve\Desktop>_
```

```
strcreator PROTO :DW

.data
    n dd 0
.data?
```

```
Microsoft Windows [Version 6.
Copyright (c) 2006 Microsoft

C:\Users\eve\Desktop>t2.exe
0
C:\Users\eve\Desktop>
```

```
strcreator PROTO :DWOR

.data
    n dd 000123
.data?
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.0.6
Copyright (c) 2006 Microsoft Co

C:\Users\eve\Desktop>t2.exe
123
C:\Users\eve\Desktop>
```

```
.data
    n dd 4444444
.data?
```

```
C:\Users\eve\Desktop>t2.exe
4444444
C:\Users\eve\Desktop>
```

```
.data
    n dd 10000000
.data?

buf db 11 dup(?)
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.
Copyright (c) 2006 Microsoft

C:\Users\eve\Desktop>t2.exe
10000000
C:\Users\eve\Desktop>_
```

```
.data
    n dd -1
.data?

buf db 11 dup(?)
```

```
Microsoft Windows [Version 6
Copyright (c) 2006 Microsoft

C:\Users\eve\Desktop>t2.exe
N is less than 0

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

```
.data
    n dd 98776655
.data?

buf db 11 dup(?)
```

```
Microsoft Windows [Version 6
Copyright (c) 2006 Microsoft

C:\Users\eve\Desktop>t2.exe
98776655
C:\Users\eve\Desktop>_
```

```
.data
    n dd 3333888
.data?
```

```
Copyright (c) 2006 Microsoft

C:\Users\eve\Desktop>t2.exe
3333888
C:\Users\eve\Desktop>
```

```
.data
    n dd -3333888
.data?
```

```
Copyright (c) 2006 Microsoft (

C:\Users\eve\Desktop>t2.exe
N is less than 0

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

```
.data
    n dd 5
.data?
```

```
C:\Users\eve\Desktop>t2.exe
5
C:\Users\eve\Desktop>
```

3. CPU, memory & I/O are the three main components of computer architecture. All three elements are connected and need each other to create a proper user experience.
- CPU – central processing unit – is used to process input data from a user and return the output. It processes the data according to code instructions and executes programs.
- Memory is a data storage unit with a limited capacity of data that can be stored (which is expressed in Bytes). Once a user enters data using input devices, the computer system stores this data in its memory unit. Users can also read information from memory, store and edit it. RAM – random-access memory – is a short-term memory where data is stored temporarily, while the processor needs it.
- I/O – input/output – is an umbrella term, unifying all kinds of devices designed either to receive data from the client to the computer (input), and to send the data from the computer to the client (output). It is a set of interfaces which allow clients communication with the computer.