```
#jobbade med Abdulkarim Dawalibi
 .data
buf: .space 64
Index: .quad 0
outbuf: .space 64
outindex: .quad 0
headMsg:    .asciz      "Start av testprogram. Skriv in 5 tal!"
endMsg:     .asciz      "Slut pa testprogram."

sum:  .quad 0
count:      .quad 0
temp: .quad 0
.text
.global     main
main:
      pushq $0
      movq  $headMsg,%rdi
      call  putText
      call  outImage
      call  inImage
      movq  $5,count
l1:
      call  getInt
      movq  %rax,temp
      cmpq  $0,%rax
      jge         l2
      call  getOutPos
      decq  %rax
      movq  %rax,%rdi
      call  setOutPos
l2:
      movq  temp,%rdx
      add   %rdx,sum
      movq  %rdx,%rdi
      call  putInt
      movq  $'+',%rdi
      call  putChar
      decq  count
      cmpq  $0,count
      jne   l1
      call  getOutPos
      decq  %rax
      movq  %rax,%rdi
      call  setOutPos
      movq  $'=',%rdi
      call  putChar
      movq  sum, %rdi
      call  putInt
      call  outImage
      movq  $12,%rsi
      movq  $buf,%rdi
      call  getText
      movq  $buf,%rdi
      call  putText
      movq  $125,%rdi
      call  putInt
      call  outImage
      movq  $endMsg,%rdi
      call  putText
      call  outImage
      popq  %rax
      ret
#_______________________________________________________
inImage:
 movq $0, Index
 movq $buf, %rdi
 movq $64, %rsi
 movq stdin, %rdx
 call   fgets
 ret
hop1:
      call inImage
      jmp  getInt
#__________________________________________________________
getInt:
      leaq  buf, %rdi
      movq  Index, %r15
      cmpb  $10, (%rdi, %r15)
      je    hop1
      movq    $0, %r10
      movq    $0, %rax
      movq    $0, %r11


signs:
      cmpb  $' ', (%rdi, %r15)
      je    inc
      cmpb  $'-', (%rdi, %r15)
      je negative
      cmpb   $'+', (%rdi, %r15)
      je positiv
      jmp convert

inc:
      incq  %r15
      jmp   signs

negative:
      movq  $1, %r11
      incq    %r15
      jmp     convert
positiv:
      movq  $0, %r11
      incq  %r15
      jmp   convert

convert:
cmpb  $'9', (%rdi, %r15)
jg    prepare
cmpb  $'0', (%rdi, %r15)
jl    prepare
movzbq (%rdi, %r15), %r10
subq  $'0',%r10
imulq   $10, %rax
addq    %r10,%rax
incq  %r15
cmpb    $' ', (%rdi, %r15)
je  prepare
```

```
jmp convert
prepare:
movq  %r15, Index
cmpq    $0, %r11
je    finish
negq    %rax

finish:
      ret


hop3:
      call inImage
      jmp  getChar
#___________________________________________________________
getChar:
    cmpb    $10, (%rdi, %r15)
    jle     hop3
      movzbq (%rdi, %r15), %r10
      incq %r15
      ret

#___________________________________________________________
getText:
leaq    buf, %rdi
movq  $0,%r9
movq  Index, %r15
cmpb    $10,(%rdi,%r15)
jle     inImage
addq  %rsi,%r15
pushq $0
decre:
cmpb  $10, (%rdi,%r15)
jle   dere
jmp  save
dere:
dec %r15
dec %rsi
jmp decre
save:
incq    %rsi
movq  %r15, Index
movq  %r15, %rax

jmp   fix

fix:
cmpq  $0,%rsi
je    fin
movzbq      (%rdi,%r15), %r9
pushq %r9
decq  %r15
decq  %rsi
jmp   fix

fin:
popq  %r9
cmpq  $0,%r9
je    returnn
```

```
movq  %r9, (%rdi,%rsi)
incq  %rsi
jmp   fin

returnn:
      ret


#_________________________________________________________

getInPos:
movq Index, %rax
ret
#_________________________________________________________
setInPos:
cmpq  $0, %rsi
jle   set_start_index
cmpb  $0, (%rdi, %r15)
jne   increase
ret
increase:
incq  %r15
decq  %rsi
jmp   setInPos
return_inpos:
ret
set_start_index:
movq  $0, %r15
ret
#______________________________________________________________



outImage:
      movq  $outbuf, %rdi
      movb  $0,%al
      call        printf
      movq $0, outbuf
      movq $0, outindex
      ret

#__________________________________________________________________

putChar:
      cmpq  $63, outindex
      jge   outImage
      movq  outindex, %r15
      leaq  outbuf,     %rdx
      movq %rdi, (%rdx, %r15)
      incq %r15
      movq  %r15, outindex
      ret
#____________________________________________________________________
putInt:
      pushq $0
      leaq outbuf, %rsi
      movq outindex, %r15

      cmpq $0, %rdi
      movq    %rdi, %rax
      jge convert_to_char
      movq $'-', (%rsi, %r15)
      negq %rdi
      incq %r15
      movq    %rdi, %rax
      convert_to_char:
            movq $10, %r10
            cqto
            divq %r10

            addq $'0', %rdx
            pushq %rdx

            cmpq $0, %rax
            jne convert_to_char

      tooutbuf:
            movq %r15, outindex
            popq %rdx
            movq %rdx, (%rsi,%r15)
            incq  %r15
            cmpq $0, %rdx
            jne tooutbuf
            ret

#__________________________________________

getOutPos:
      movq  outindex, %rax
      ret
#_______________________________________
setOutPos:
      cmpq  $0, %rdi
      jge   out_range
      movq $0, outindex
      ret
      out_range:
            cmpq  $63, %rdi
            jle   return_outpos
            movq  $63, %rdi

      return_outpos:
            movq %rdi, outindex
            ret
#_______________________________________________
putText:
      movq  outindex, %r10
      leaq  outbuf,     %rcx
      movq $10,(%rcx,%r10)
      incq %r10
      move_to_outbuf:
            cmpb  $10, (%rdi)
            jle   returnput_text
            movzbq (%rdi),%r11
            movq %r11, (%rcx, %r10)
            incq %rdi
            incq %r10
            jmp move_to_outbuf
```

```
returnput_text:
      movq $10,(%rcx,%r10)
      incq %r10
      movq  %r10, outindex
      movq  outindex, %r10
      ret
```