

```
typedef unsigned int uint;

uint f(uint x[][20], uint n)
{
    uint s = 0;
    for (uint i=0; i<n; i++)
        for (uint j=0; j<20; j++)
            s += x[i][j];
    return s;
}
```

32-bitový assembler: -march=athlon -O2 -m32

```
f:      pushl %ebp                                cmpl    %ebx, %esi
        xorl %eax, %eax                       ja      .L7
        movl %esp, %ebp                       .L3:    popl    %ebx
        pushl %esi                            popl    %esi
        movl 12(%ebp), %esi                    leave
        movl 8(%ebp), %ecx                     ret
        pushl %ebx
        xorl %ebx, %ebx
        testl %esi, %esi
        je   .L3
.L7:    xorl %edx, %edx
.L4:    addl (%ecx,%edx,4), %eax
        incl %edx
        cmpl $20, %edx
        jne .L4
        incl %ebx
        addl $80, %ecx
```

32-bitový assembler: -fomit-frame-pointer

```
f:    pushl %esi                cmpl %ebx, %esi
      xorl %eax, %eax        ja   .L7
      pushl %ebx             .L3: popl %ebx
      movl 16(%esp), %esi    popl %esi
      xorl %ebx, %ebx        ret
      movl 12(%esp), %ecx
      testl %esi, %esi
      je   .L3
.L7:  xorl %edx, %edx
.L4:  addl (%ecx,%edx,4), %eax
      incl %edx
      cmpl $20, %edx
      jne .L4
      incl %ebx
      addl $80, %ecx
```

64-bitový assembler, starší GCC: -march=k8

```
f:    testl %esi, %esi
      je    .L10
      xorl %eax, %eax
      xorl %ecx, %ecx
.L5:  xorl %edx, %edx
.L4:  addl  (%rdi,%rdx), %eax
      addq $4, %rdx
      cmpq $80, %rdx
      jne  .L4
      addl $1, %ecx
      addq $80, %rdi
      cmpl %ecx, %esi
      ja   .L5
      ret
.L10: xorl %eax, %eax
      ret
```

AMD64, novější GCC: -march=znver2

```
testl %esi, %esi
je .L5
leal -1(%rsi), %eax
leaq 80(%rdi), %rdx
xorl %r8d, %r8d
leaq (%rax,%rax,4), %rax
salq $4, %rax
leaq 160(%rdi,%rax), %rcx
.L4: leaq -80(%rdx), %rax
.L3: addl (%rax), %r8d
addq $4, %rax
cmpq %rdx, %rax
jne .L3
addq $80, %rdx
cmpq %rcx, %rdx
jne .L4
movl %r8d, %eax
ret
.L5: xorl %r8d, %r8d
movl %r8d, %eax
ret
```

Rozbalená vnitřní smyčka: -funroll-loops

```
.L4: leaq  -80(%rcx), %rdi
.L3: addl  (%rdi), %eax
      addq  $40, %rdi
      addl  -36(%rdi), %eax
      addl  -32(%rdi), %eax
      addl  -28(%rdi), %eax
      addl  -24(%rdi), %eax
      addl  -20(%rdi), %eax
      addl  -16(%rdi), %eax
      addl  -12(%rdi), %eax
      addl  -8(%rdi), %eax
      addl  -4(%rdi), %eax
      cmpq  %rcx, %rdi
      jne  .L3
```