

```

1. gcc -S 01_simple1.c
int main(void)
{
    return 42;
}
gcc -o wykonywalny 01_simple1.s
    .file    "01_simple1.c"
    .text
.globl main
    .type   main, @function
main:
.LFB2:
    pushq   %rbp
.LCFI0:
    movq    %rsp, %rbp
.LCFI1:
    movl    $42, %eax
    leave
    ret
.LFE2:
    .size   main, .-main
    .section      .eh_frame,"a",@progbits
.Lframe1:
    .long   .LECIE1-.LSCIE1
.LSCIE1:
    .long   0x0
    .byte   0x1
    .string "zR"
    .uleb128 0x1           #unsigned little endian base
    .sleb128 -8           #signed little endian base
    .byte   0x10
    .uleb128 0x1
    .byte   0x3
    .byte   0xc
    .uleb128 0x7
    .uleb128 0x8
    .byte   0x90
    .uleb128 0x1
    .align 8
.LECIE1:
.LSFDE1:
    .long   .LEFDE1-.LASFDE1
.LASFDE1:
    .long   .LASFDE1-.Lframe1
    .long   .LFB2
    .long   .LFE2-.LFB2
    .uleb128 0x0           #unsigned little endian base
    .byte   0x4
    .long   .LCFI0-.LFB2
    .byte   0xe
    .uleb128 0x10
    .byte   0x86
    .uleb128 0x2
    .byte   0x4
    .long   .LCFI1-.LCFI0
    .byte   0xd
    .uleb128 0x6
    .align 8
.LEFDE1:
    .ident  "GCC: (GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4)"
    .section      .note.GNU-stack,"",@progbits

```

Intel (vm.py)	AT&T
Wartość bezwzględna: 42 (np. push 42)	\$42 (np. pushl \$42)
Rejestry: R0, EAX, EBX (np. MOV EAX, 42)	%EAX, %ebp (np. MOVL \$42, %eax)
Kolejność: cel, źródło (np. MOV EBX, 42)	Źródło, cel (np. MOVL \$42, %ebx)
Skoki: 999 (np. JMP 999)	*999 (np. JMP *999)
Rozmiar argumentów: byte ptr, word ptr, qword ptr (np. mov al, byte ptr \$42)	b, w, l na końcu mnemonika (np. movb \$42, %al)

```

2.
int main(void)
{
    int x=17+25;
    return x;
}
    .file      "01_simple1.c"
    .text
.globl main
    .type     main, @function
main:
.LFB2:
    pushq   %rbp
.LCFI0:
    movq    %rsp, %rbp
.LCFI1:
    movl    $42, -4(%rbp)
    movl    -4(%rbp), %eax
    leave
    ret
.LFE2:
    .size   main, .-main
    .section      .eh_frame,"a",@progbits
.Lframe1:
    .long   .LECIE1-.LSCIE1
.LSCIE1:
    .long   0x0
    .byte   0x1
    .string "zR"
    .uleb128 0x1           #unsigned little endian base
    .sleb128 -8           #signed little endian base
    .byte   0x10
    .uleb128 0x1
    .byte   0x3
    .byte   0xc
    .uleb128 0x7
    .uleb128 0x8
    .byte   0x90
    .uleb128 0x1
    .align  8
.LECIE1:
.LSFDE1:
    .long   .LEFDE1-.LASFDE1
.LASFDE1:
    .long   .LASFDE1-.Lframe1
    .long   .LFB2
    .long   .LFE2-.LFB2
    .uleb128 0x0           #unsigned little endian base
    .byte   0x4
    .long   .LCFI0-.LFB2
    .byte   0xe
    .uleb128 0x10
    .byte   0x86
    .uleb128 0x2
    .byte   0x4
    .long   .LCFI1-.LCFI0
    .byte   0xd
    .uleb128 0x6
    .align  8
.LEFDE1:
    .ident  "GCC: (GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4)"
    .section      .note.GNU-stack,"",@progbits

```

```

3.
int main(void)
{
    int x=17;
    x+=25;
    return x;
}

```

```

.LCFI1:
    movl    $17, -4(%rbp)
    addl    $25, -4(%rbp)
    movl    -4(%rbp), %eax
    movl    -4(%rbp), %eax
    leave
    ret

```

```

4.
int main(void)
{
    int x=17;
    x*=11;
    return x;
}

```

```

.LCFI1:
    movl    $17, -4(%rbp)
    movl    -4(%rbp), %edx
    movl    %edx, %eax
    sall    $2, %eax
    addl    %edx, %eax
    addl    %eax, %eax
    addl    %edx, %eax
    movl    %eax, -4(%rbp)
    movl    -4(%rbp), %eax
    leave
    ret

```

```

5.
int main(void)
{
    int x=17;
    x*=11;
    return x;
}

```

```

.LCFI1:
    movl    $17, -4(%rbp)
    movl    -4(%rbp), %edx
    movl    %edx, %eax
    sall    $2, %eax
    addl    %edx, %eax
    addl    %eax, %eax
    addl    %edx, %eax
    movl    %eax, -4(%rbp)
    movl    -4(%rbp), %eax
    leave
    ret

```

```
6.  
int main(void)  
{  
    int x=13;  
    x*=317;  
    return x;  
}
```

```
.LCFI1:  
    movl    $13, -4(%rbp)  
    movl    -4(%rbp), %eax  
    imull   $317, %eax, %eax  
    movl    %eax, -4(%rbp)  
    movl    -4(%rbp), %eax  
    leave  
    ret
```

```

7.
int main(void)
{
    int a, b;
    int x;
    scanf("%d\n", &a);
    scanf("%d\n", &b);
    x = a+b;
    return x;
}

```

```

        .file   "04_scanf1.c"
        .section .rodata
.LC0:
        .string "%d\n"
        .text
.globl main
        .type   main, @function
main:
.LFB2:
        pushq   %rbp
.LCFI0:
        movq    %rsp, %rbp
.LCFI1:
        subq    $16, %rsp
.LCFI2:
        leaq   -4(%rbp), %rsi
        movl   $.LC0, %edi
        movl   $0, %eax
        call   scanf
        leaq   -8(%rbp), %rsi
        movl   $.LC0, %edi
        movl   $0, %eax
        call   scanf
        movl   -4(%rbp), %edx
        movl   -8(%rbp), %eax
        leal   (%rdx,%rax), %eax
        movl   %eax, -12(%rbp)
        movl   -12(%rbp), %eax
        leave
        ret
.LFE2:
        .size   main, .-main
        .section .eh_frame,"a",@progbits
.Lframe1:
        .long   .LECIE1-.LSCIE1
.LSCIE1:
        .long   0x0
        .byte   0x1
        .string "zR"
        .uleb128 0x1
        .sleb128 -8
        .byte   0x10
        .uleb128 0x1
        .byte   0x3
        .byte   0xc
        .uleb128 0x7
        .uleb128 0x8
        .byte   0x90
        .uleb128 0x1
        .align 8

```

```
.LECIE1:
.LSFDE1:
    .long    .LEFDE1-.LASFDE1
.LASFDE1:
    .long    .LASFDE1-.Lframe1
    .long    .LFB2
    .long    .LFE2-.LFB2
    .uleb128 0x0
    .byte    0x4
    .long    .LCFI0-.LFB2
    .byte    0xe
    .uleb128 0x10
    .byte    0x86
    .uleb128 0x2
    .byte    0x4
    .long    .LCFI1-.LCFI0
    .byte    0xd
    .uleb128 0x6
    .align   8
.LEFDE1:
    .ident   "GCC: (GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4)"
    .section .note.GNU-stack,"",@progbits
```