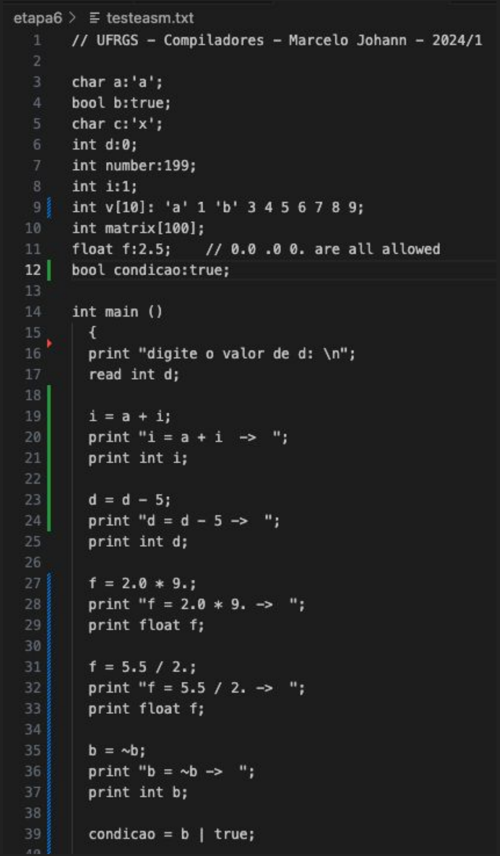
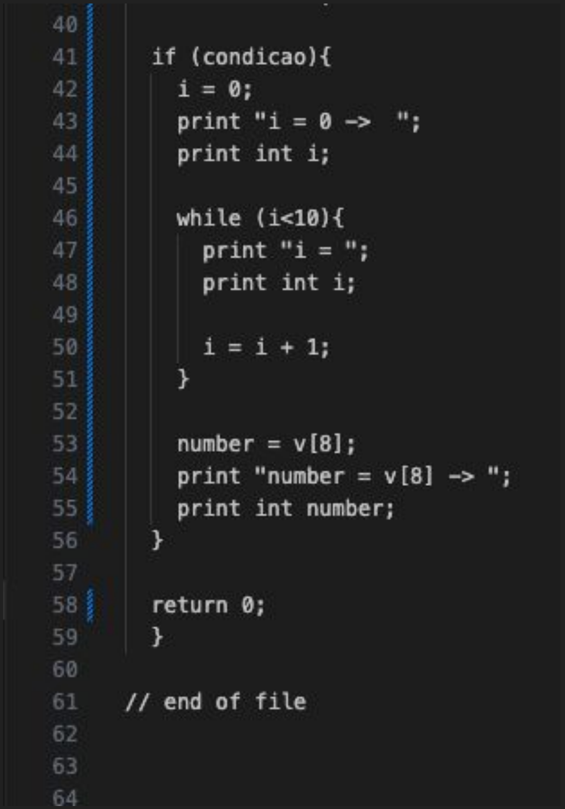
Projeto de Compilador
Etapa 6 - Assembly

Matheus Lima

Programa de exemplo

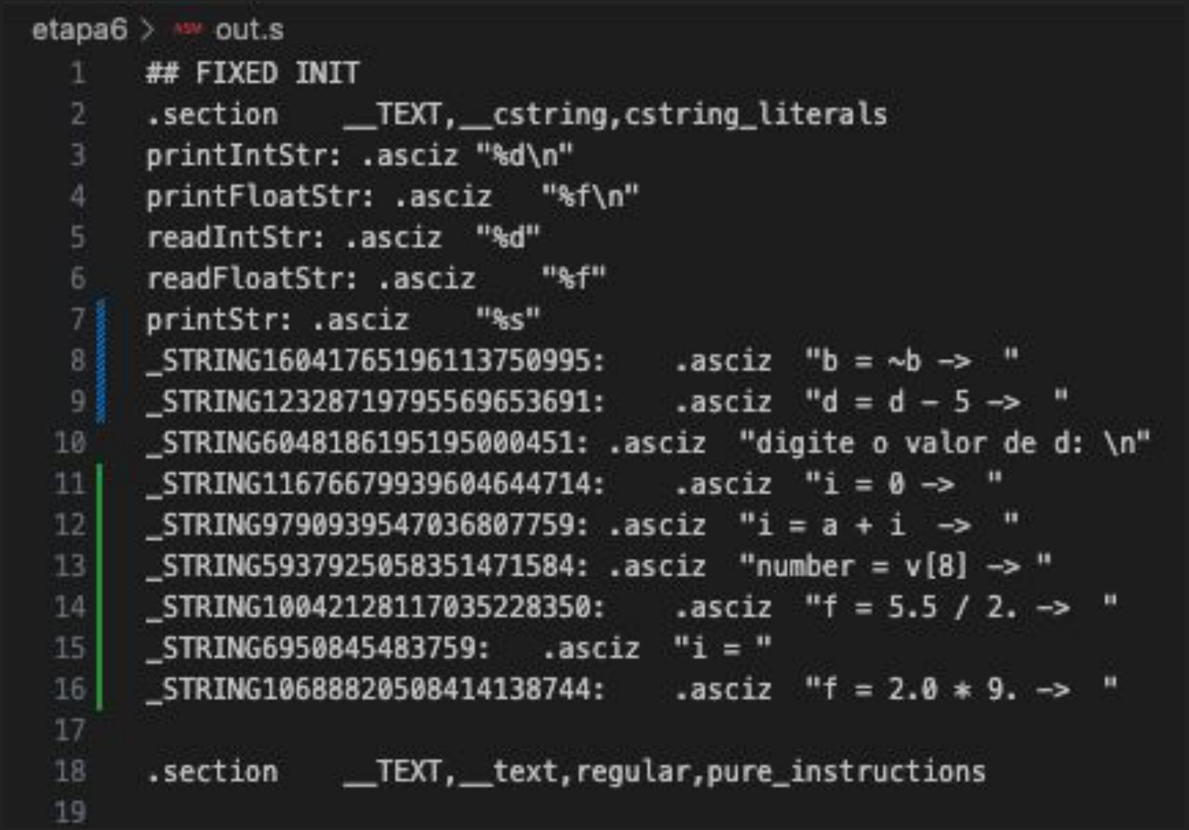


```
// UFRGS - Compiladores - Marcelo Johann - 2024/1

char a:'a';
bool b:true;
char c:'x';
int d:0;
int number:199;
int i:1;
int v[10]: 'a' 1 'b' 3 4 5 6 7 8 9;
int matrix[100];
float f:2.5;    // 0.0 .0 0. are all allowed
bool condicao:true;

int main ()
  {
  print "digite o valor de d: \n";
  read int d;

  i = a + i;
  print "i = a + i  ->  ";
  print int i;

  d = d - 5;
  print "d = d - 5 ->  ";
  print int d;

  f = 2.0 * 9.;
  print "f = 2.0 * 9. ->  ";
  print float f;

  f = 5.5 / 2.;
  print "f = 5.5 / 2. ->  ";
  print float f;

  b = ~b;
  print "b = ~b ->  ";
  print int b;

  condicao = b | true;
```



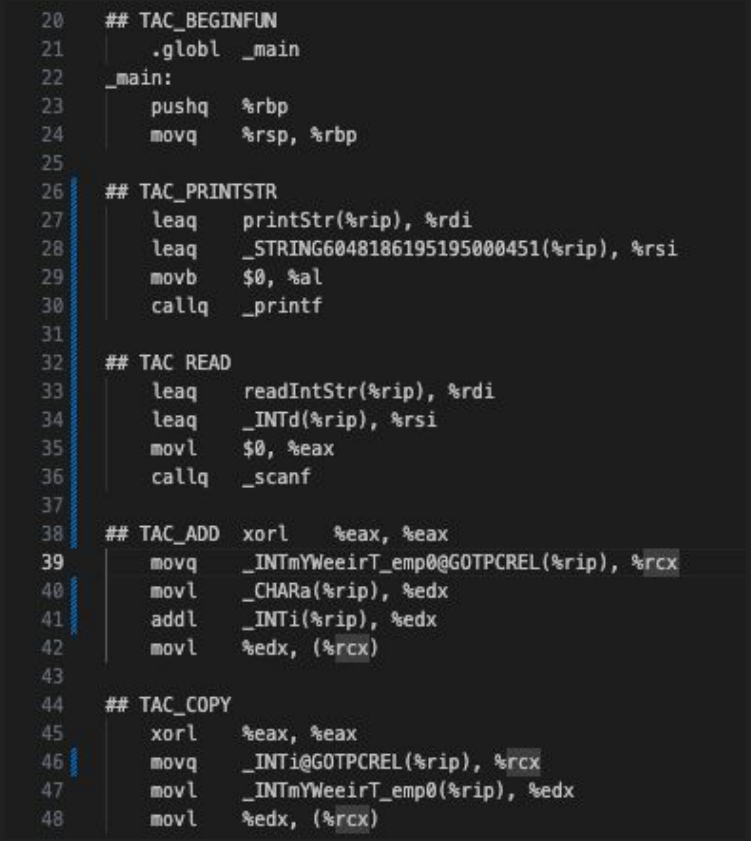
```

  if (condicao){
    i = 0;
    print "i = 0 ->  ";
    print int i;

    while (i<10){
      print "i = ";
      print int i;

      i = i + 1;
    }

    number = v[8];
    print "number = v[8] -> ";
    print int number;
  }

  return 0;
  }

// end of file



```

Assembly gerado - Strings

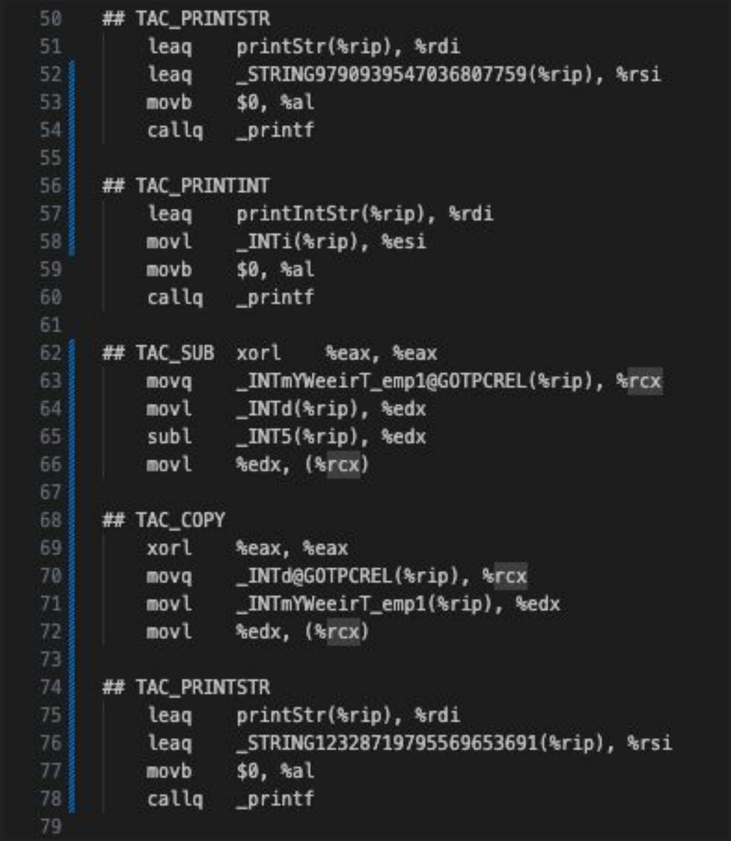


```
## FIXED INIT
.section    __TEXT,__cstring,cstring_literals
printIntStr: .asciz "%d\n"
printFloatStr: .asciz   "%f\n"
readIntStr: .asciz  "%d"
readFloatStr: .asciz    "%f"
printStr: .asciz    "%s"
_STRING16041765196113750995:    .asciz  "b = ~b ->  "
_STRING12328719795569653691:    .asciz  "d = d - 5 ->  "
_STRING6048186195195000451: .asciz  "digite o valor de d: \n"
_STRING11676679939604644714:    .asciz  "i = 0 ->  "
_STRING9790939547036807759: .asciz  "i = a + i  ->  "
_STRING5937925058351471584: .asciz  "number = v[8] -> "
_STRING10042128117035228350:    .asciz  "f = 5.5 / 2. ->  "
_STRING6950845483759:   .asciz  "i = "
_STRING10688820508414138744:    .asciz  "f = 2.0 * 9. ->  "

.section    __TEXT,__text,regular,pure_instructions

```

Assembly gerado - Código

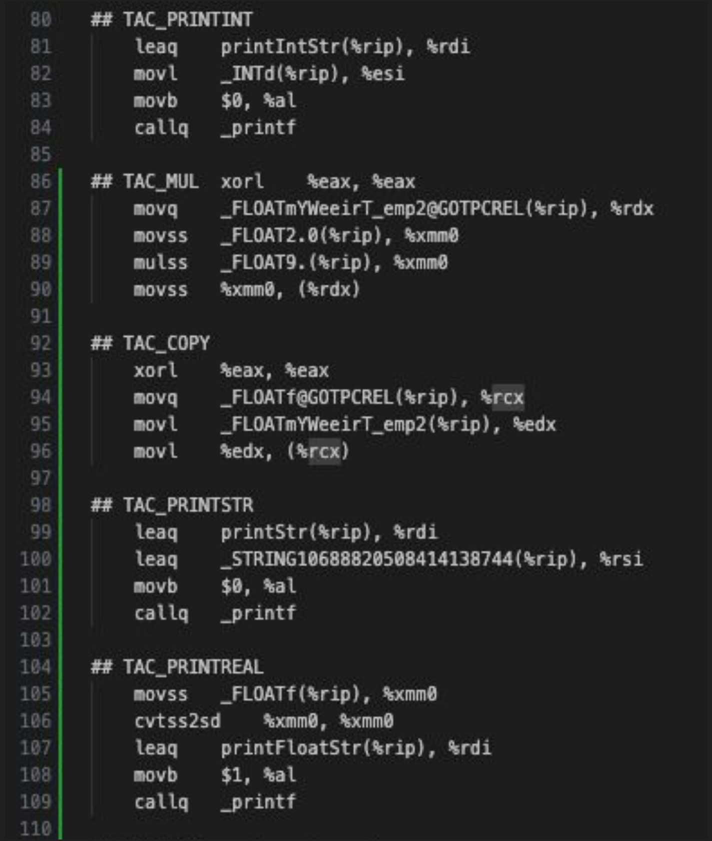


```
## TAC_BEGINFUN
    .globl  _main
_main:
    pushq   %rbp
    movq    %rsp, %rbp

## TAC_PRINTSTR
    leaq    printStr(%rip), %rdi
    leaq    _STRING6048186195195000451(%rip), %rsi
    movb    $0, %al
    callq   _printf

## TAC READ
    leaq    readIntStr(%rip), %rdi
    leaq    _INTd(%rip), %rsi
    movl    $0, %eax
    callq   _scanf

## TAC_ADD  xorl    %eax, %eax
    movq    _INTmYWeeirT_emp0@GOTPCREL(%rip), %rcx
    movl    _CHARa(%rip), %edx
    addl    _INTi(%rip), %edx
    movl    %edx, (%rcx)

## TAC_COPY
    xorl    %eax, %eax
    movq    _INTi@GOTPCREL(%rip), %rcx
    movl    _INTmYWeeirT_emp0(%rip), %edx
    movl    %edx, (%rcx)
```

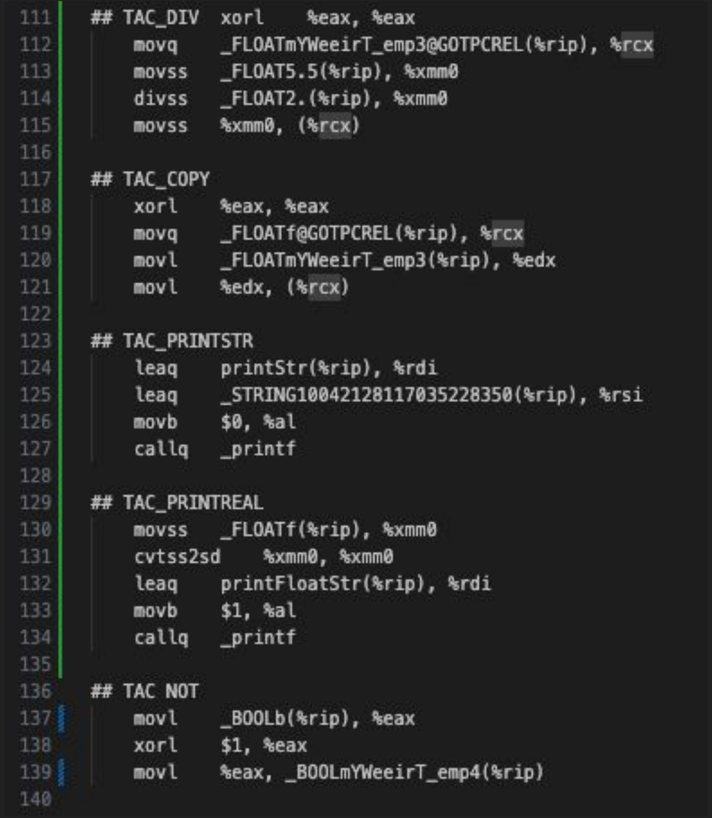


```
## TAC_PRINTSTR
    leaq    printStr(%rip), %rdi
    leaq    _STRING9790939547036807759(%rip), %rsi
    movb    $0, %al
    callq   _printf

## TAC_PRINTINT
    leaq    printIntStr(%rip), %rdi
    movl    _INTi(%rip), %esi
    movb    $0, %al
    callq   _printf

## TAC_SUB  xorl    %eax, %eax
    movq    _INTmYWeeirT_emp1@GOTPCREL(%rip), %rcx
    movl    _INTd(%rip), %edx
    subl    _INT5(%rip), %edx
    movl    %edx, (%rcx)

## TAC_COPY
    xorl    %eax, %eax
    movq    _INTd@GOTPCREL(%rip), %rcx
    movl    _INTmYWeeirT_emp1(%rip), %edx
    movl    %edx, (%rcx)

## TAC_PRINTSTR
    leaq    printStr(%rip), %rdi
    leaq    _STRING12328719795569653691(%rip), %rsi
    movb    $0, %al
    callq   _printf
```

Assembly gerado - Código

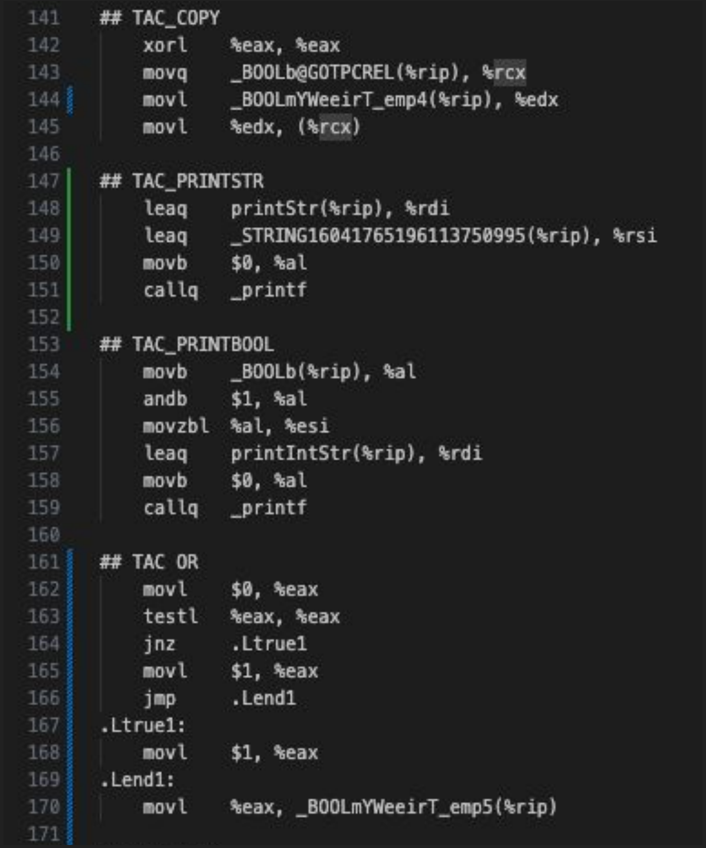


```
## TAC_PRINTINT
    leaq    printIntStr(%rip), %rdi
    movl    _INTd(%rip), %esi
    movb    $0, %al
    callq   _printf

## TAC_MUL  xorl    %eax, %eax
    movq    _FLOATmYWeeirT_emp2@GOTPCREL(%rip), %rdx
    movss   _FLOAT2.0(%rip), %xmm0
    mulss   _FLOAT9.(%rip), %xmm0
    movss   %xmm0, (%rdx)

## TAC_COPY
    xorl    %eax, %eax
    movq    _FLOATf@GOTPCREL(%rip), %rcx
    movl    _FLOATmYWeeirT_emp2(%rip), %edx
    movl    %edx, (%rcx)

## TAC_PRINTSTR
    leaq    printStr(%rip), %rdi
    leaq    _STRING10688820508414138744(%rip), %rsi
    movb    $0, %al
    callq   _printf

## TAC_PRINTREAL
    movss   _FLOATf(%rip), %xmm0
    cvtss2sd    %xmm0, %xmm0
    leaq    printFloatStr(%rip), %rdi
    movb    $1, %al
    callq   _printf

```

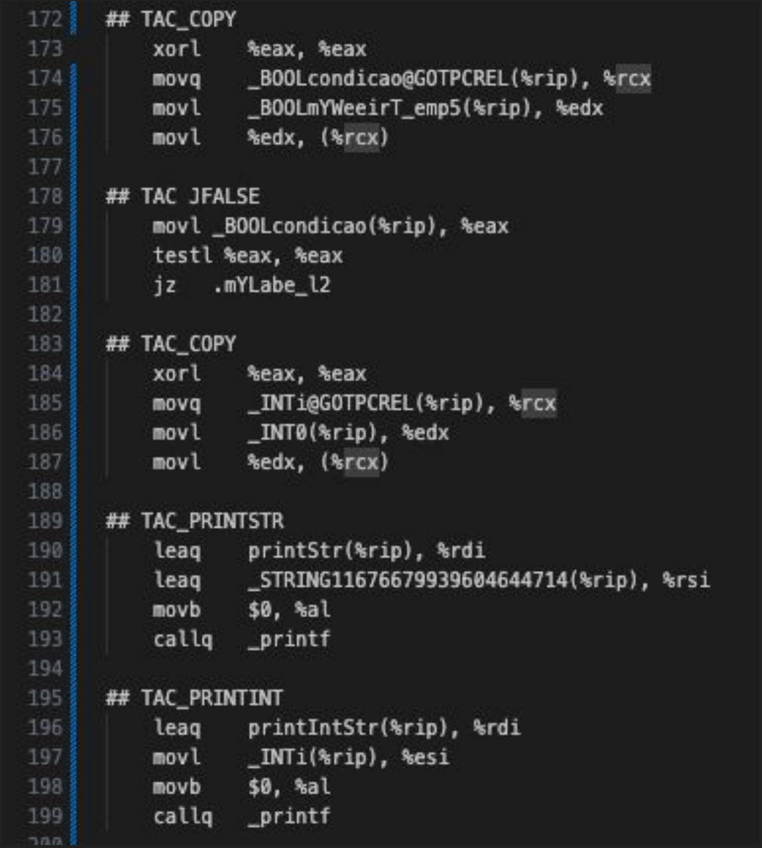


```
## TAC_DIV  xorl    %eax, %eax
    movq    _FLOATmYWeeirT_emp3@GOTPCREL(%rip), %rcx
    movss   _FLOAT5.5(%rip), %xmm0
    divss   _FLOAT2.(%rip), %xmm0
    movss   %xmm0, (%rcx)

## TAC_COPY
    xorl    %eax, %eax
    movq    _FLOATf@GOTPCREL(%rip), %rcx
    movl    _FLOATmYWeeirT_emp3(%rip), %edx
    movl    %edx, (%rcx)

## TAC_PRINTSTR
    leaq    printStr(%rip), %rdi
    leaq    _STRING10042128117035228350(%rip), %rsi
    movb    $0, %al
    callq   _printf

## TAC_PRINTREAL
    movss   _FLOATf(%rip), %xmm0
    cvtss2sd    %xmm0, %xmm0
    leaq    printFloatStr(%rip), %rdi
    movb    $1, %al
    callq   _printf

## TAC NOT
    movl    _BOOLb(%rip), %eax
    xorl    $1, %eax
    movl    %eax, _BOOLmYWeeirT_emp4(%rip)

```

Assembly gerado - Código

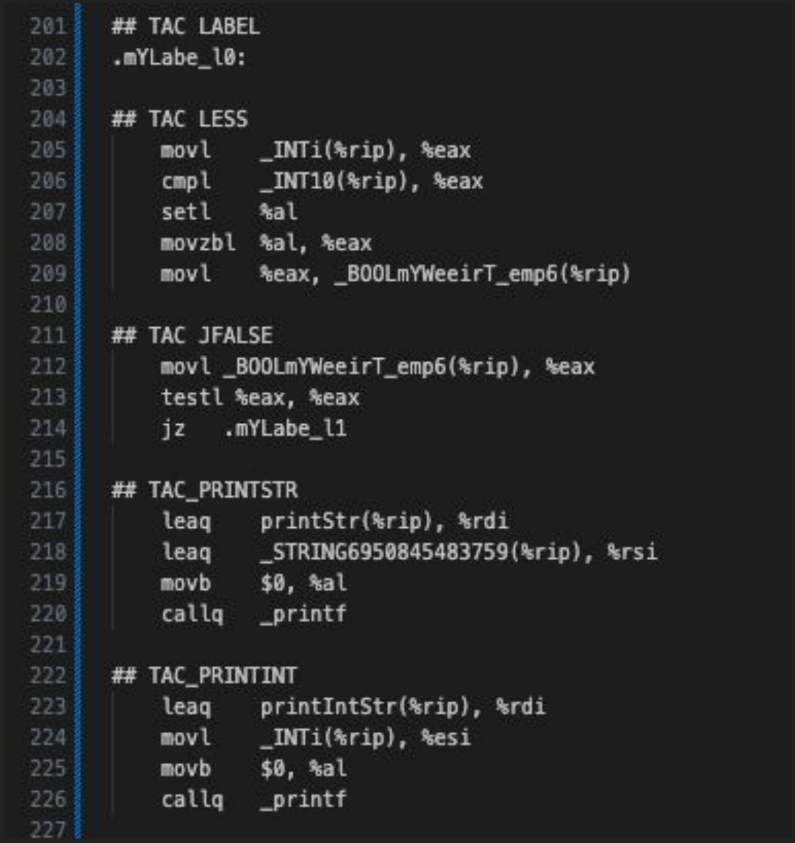


```
## TAC_COPY
    xorl    %eax, %eax
    movq    _BOOLb@GOTPCREL(%rip), %rcx
    movl    _BOOLmYWeeirT_emp4(%rip), %edx
    movl    %edx, (%rcx)

## TAC_PRINTSTR
    leaq    printStr(%rip), %rdi
    leaq    _STRING16041765196113750995(%rip), %rsi
    movb    $0, %al
    callq   _printf

## TAC_PRINTBOOL
    movb    _BOOLb(%rip), %al
    andb    $1, %al
    movzbl  %al, %esi
    leaq    printIntStr(%rip), %rdi
    movb    $0, %al
    callq   _printf

## TAC OR
    movl    $0, %eax
    testl   %eax, %eax
    jnz     .Ltrue1
    movl    $1, %eax
    jmp     .Lend1
.Ltrue1:
    movl    $1, %eax
.Lend1:
    movl    %eax, _BOOLmYWeeirT_emp5(%rip)
```

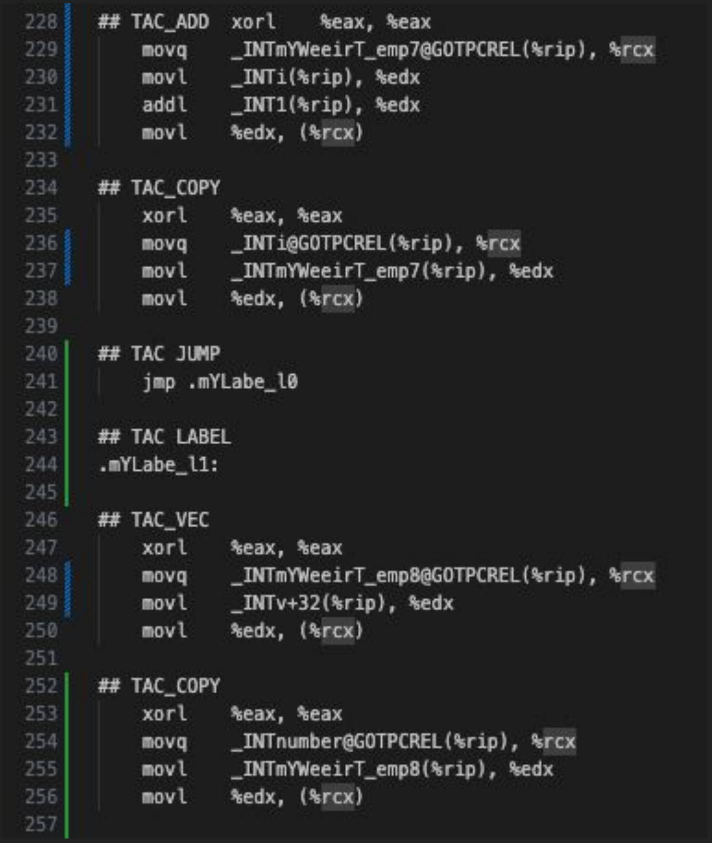


```
## TAC_COPY
    xorl    %eax, %eax
    movq    _BOOLcondicao@GOTPCREL(%rip), %rcx
    movl    _BOOLmYWeeirT_emp5(%rip), %edx
    movl    %edx, (%rcx)

## TAC JFALSE
    movl _BOOLcondicao(%rip), %eax
    testl %eax, %eax
    jz   .mYLabe_l2

## TAC_COPY
    xorl    %eax, %eax
    movq    _INTi@GOTPCREL(%rip), %rcx
    movl    _INT0(%rip), %edx
    movl    %edx, (%rcx)

## TAC_PRINTSTR
    leaq    printStr(%rip), %rdi
    leaq    _STRING11676679939604644714(%rip), %rsi
    movb    $0, %al
    callq   _printf

## TAC_PRINTINT
    leaq    printIntStr(%rip), %rdi
    movl    _INTi(%rip), %esi
    movb    $0, %al
    callq   _printf
```

Assembly gerado - Código

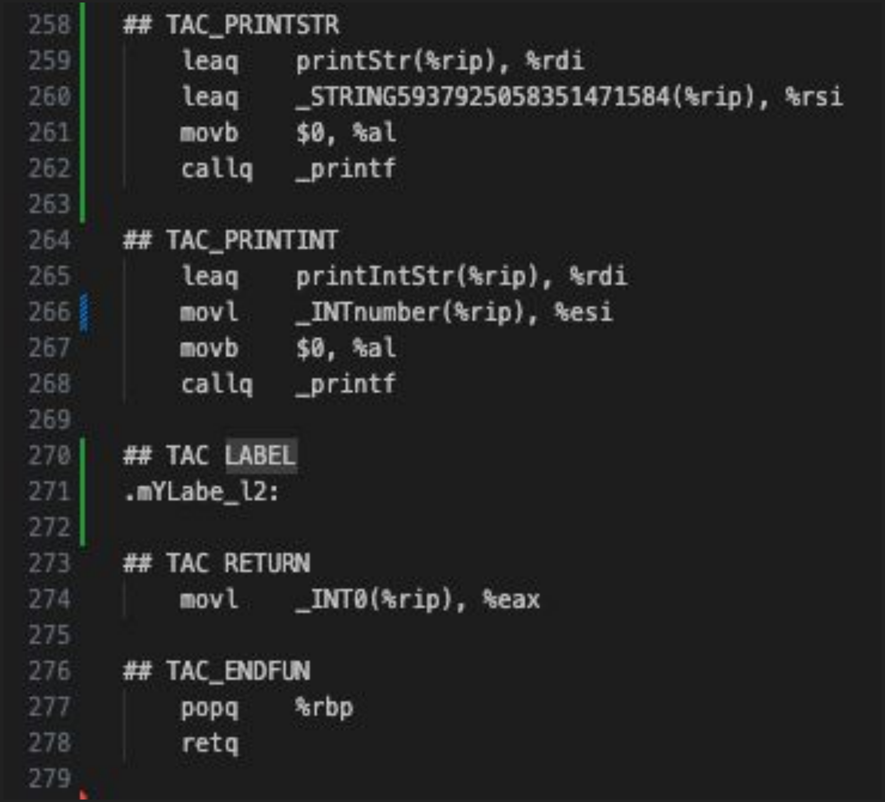


```
## TAC LABEL
.mYLabe_l0:

## TAC LESS
    movl    _INTi(%rip), %eax
    cmpl    _INT10(%rip), %eax
    setl    %al
    movzbl  %al, %eax
    movl    %eax, _BOOLmYWeeirT_emp6(%rip)

## TAC JFALSE
    movl _BOOLmYWeeirT_emp6(%rip), %eax
    testl %eax, %eax
    jz   .mYLabe_l1

## TAC_PRINTSTR
    leaq    printStr(%rip), %rdi
    leaq    _STRING6950845483759(%rip), %rsi
    movb    $0, %al
    callq   _printf

## TAC_PRINTINT
    leaq    printIntStr(%rip), %rdi
    movl    _INTi(%rip), %esi
    movb    $0, %al
    callq   _printf

```



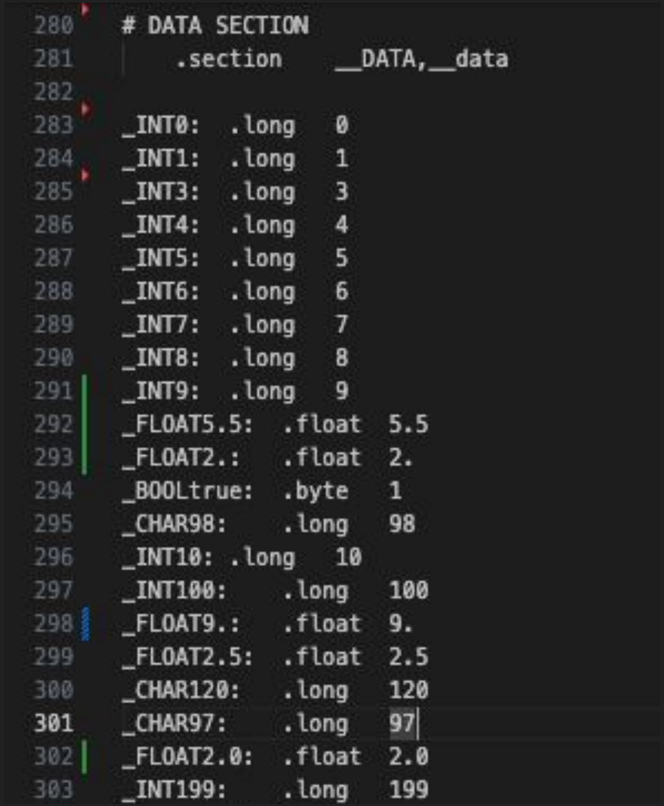
```
## TAC_ADD  xorl    %eax, %eax
    movq    _INTmYWeeirT_emp7@GOTPCREL(%rip), %rcx
    movl    _INTi(%rip), %edx
    addl    _INT1(%rip), %edx
    movl    %edx, (%rcx)

## TAC_COPY
    xorl    %eax, %eax
    movq    _INTi@GOTPCREL(%rip), %rcx
    movl    _INTmYWeeirT_emp7(%rip), %edx
    movl    %edx, (%rcx)

## TAC JUMP
    jmp .mYLabe_l0

## TAC LABEL
.mYLabe_l1:

## TAC_VEC
    xorl    %eax, %eax
    movq    _INTmYWeeirT_emp8@GOTPCREL(%rip), %rcx
    movl    _INTv+32(%rip), %edx
    movl    %edx, (%rcx)

## TAC_COPY
    xorl    %eax, %eax
    movq    _INTnumber@GOTPCREL(%rip), %rcx
    movl    _INTmYWeeirT_emp8(%rip), %edx
    movl    %edx, (%rcx)

```

Assembly gerado - Código

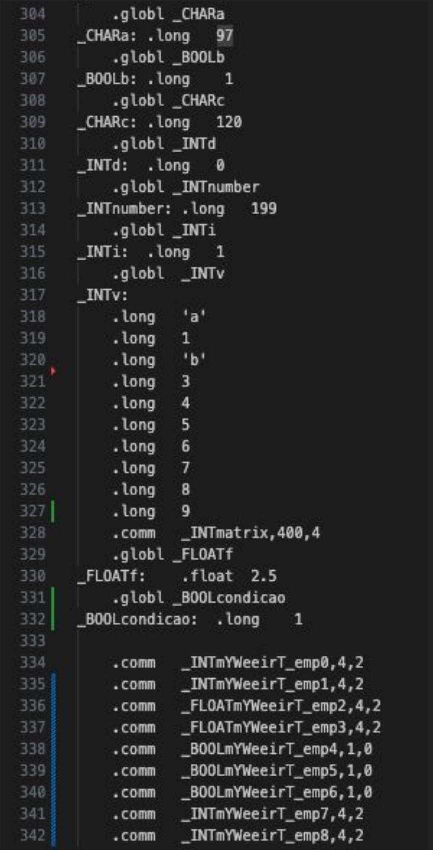


```
## TAC_PRINTSTR
    leaq    printStr(%rip), %rdi
    leaq    _STRING5937925058351471584(%rip), %rsi
    movb    $0, %al
    callq   _printf

## TAC_PRINTINT
    leaq    printIntStr(%rip), %rdi
    movl    _INTnumber(%rip), %esi
    movb    $0, %al
    callq   _printf

## TAC LABEL
.mYLabe_l2:

## TAC RETURN
    movl    _INT0(%rip), %eax

## TAC_ENDFUN
    popq    %rbp
    retq
```

Assembly gerado - Data Section

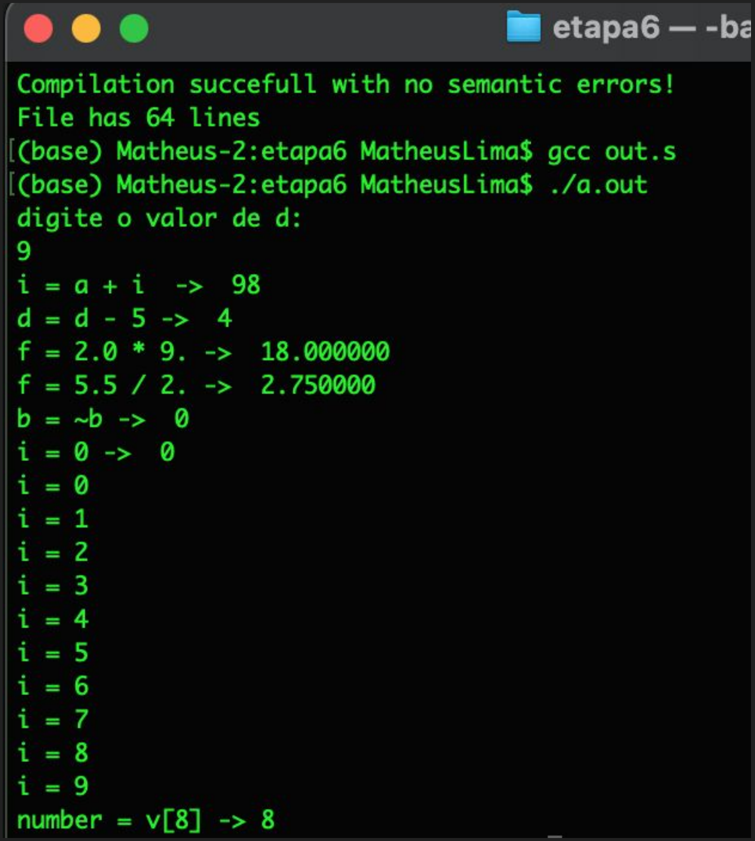


```
# DATA SECTION
    .section    __DATA,__data

_INT0:  .long   0
_INT1:  .long   1
_INT3:  .long   3
_INT4:  .long   4
_INT5:  .long   5
_INT6:  .long   6
_INT7:  .long   7
_INT8:  .long   8
_INT9:  .long   9
_FLOAT5.5:  .float  5.5
_FLOAT2.:   .float  2.
_BOOLtrue:  .byte   1
_CHAR98:    .long   98
_INT10: .long   10
_INT100:    .long   100
_FLOAT9.:   .float  9.
_FLOAT2.5:  .float  2.5
_CHAR120:   .long   120
_CHAR97:    .long   97
_FLOAT2.0:  .float  2.0
_INT199:    .long   199
```



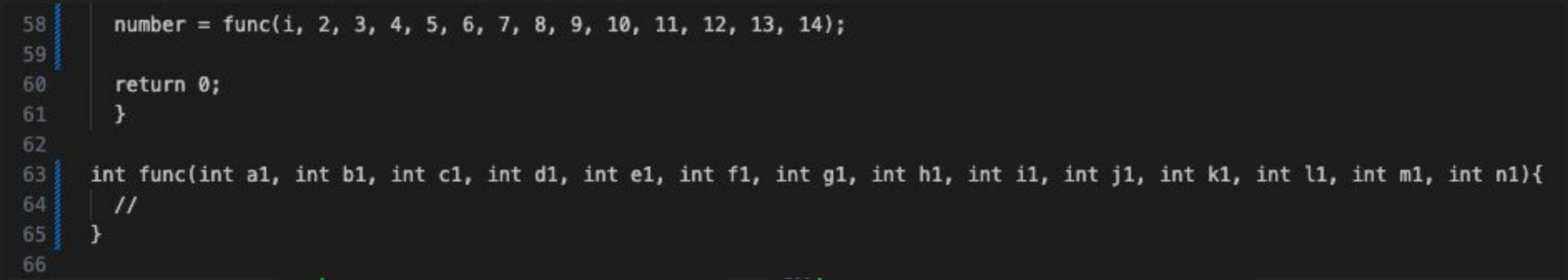
```
    .globl _CHARa
_CHARa: .long   97
    .globl _BOOLb
_BOOLb: .long    1
    .globl _CHARc
_CHARc: .long   120
    .globl _INTd
_INTd:  .long   0
    .globl _INTnumber
_INTnumber: .long   199
    .globl _INTi
_INTi:  .long   1
    .globl  _INTv
_INTv:
    .long   'a'
    .long   1
    .long   'b'
    .long   3
    .long   4
    .long   5
    .long   6
    .long   7
    .long   8
    .long   9
    .comm   _INTmatrix,400,4
    .globl _FLOATf
_FLOATf:    .float  2.5
    .globl _BOOLcondicao
_BOOLcondicao:  .long    1

    .comm   _INTmYWeeirT_emp0,4,2
    .comm   _INTmYWeeirT_emp1,4,2
    .comm   _FLOATmYWeeirT_emp2,4,2
    .comm   _FLOATmYWeeirT_emp3,4,2
    .comm   _BOOLmYWeeirT_emp4,1,0
    .comm   _BOOLmYWeeirT_emp5,1,0
    .comm   _BOOLmYWeeirT_emp6,1,0
    .comm   _INTmYWeeirT_emp7,4,2
    .comm   _INTmYWeeirT_emp8,4,2
```

Saída

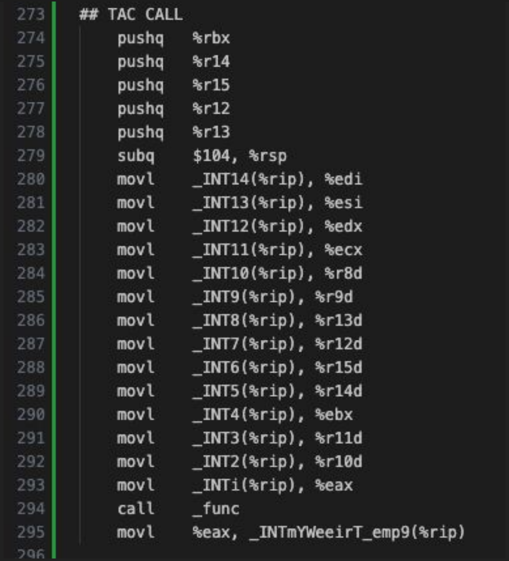


```
Compilation succefull with no semantic errors!
File has 64 lines
(base) Matheus-2:etapa6 MatheusLima$ gcc out.s
(base) Matheus-2:etapa6 MatheusLima$ ./a.out
digite o valor de d:
9
i = a + i  ->  98
d = d - 5 ->  4
f = 2.0 * 9. ->  18.000000
f = 5.5 / 2. ->  2.750000
b = ~b ->  0
i = 0 ->  0
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
number = v[8] -> 8
```

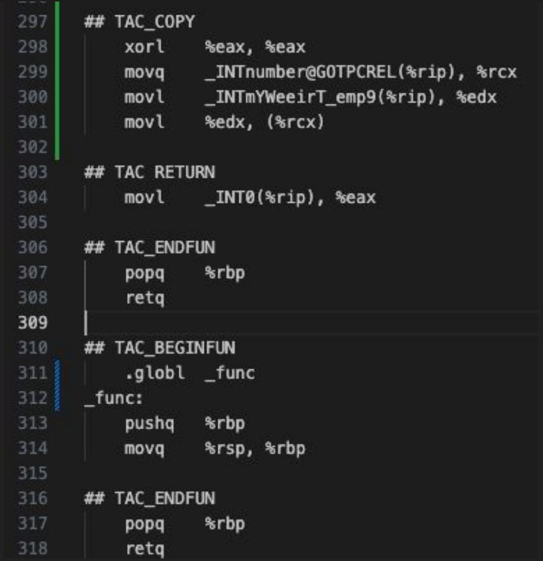
Caso incompleto - Funções



```
    number = func(i, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14);

    return 0;
    }

int func(int a1, int b1, int c1, int d1, int e1, int f1, int g1, int h1, int i1, int j1, int k1, int l1, int m1, int n1){
    //
}
```



```
## TAC CALL
    pushq   %rbx
    pushq   %r14
    pushq   %r15
    pushq   %r12
    pushq   %r13
    subq    $104, %rsp
    movl    _INT14(%rip), %edi
    movl    _INT13(%rip), %esi
    movl    _INT12(%rip), %edx
    movl    _INT11(%rip), %ecx
    movl    _INT10(%rip), %r8d
    movl    _INT9(%rip), %r9d
    movl    _INT8(%rip), %r13d
    movl    _INT7(%rip), %r12d
    movl    _INT6(%rip), %r15d
    movl    _INT5(%rip), %r14d
    movl    _INT4(%rip), %ebx
    movl    _INT3(%rip), %r11d
    movl    _INT2(%rip), %r10d
    movl    _INTi(%rip), %eax
    call    _func
    movl    %eax, _INTmYWeeirT_emp9(%rip)
```



```
## TAC_COPY
    xorl    %eax, %eax
    movq    _INTnumber@GOTPCREL(%rip), %rcx
    movl    _INTmYWeeirT_emp9(%rip), %edx
    movl    %edx, (%rcx)

## TAC RETURN
    movl    _INT0(%rip), %eax

## TAC_ENDFUN
    popq    %rbp
    retq

## TAC_BEGINFUN
    .globl  _func
_func:
    pushq   %rbp
    movq    %rsp, %rbp

## TAC_ENDFUN
    popq    %rbp
    retq
```