Practica 1

DOCENTE	CARRERA	CURSO
MSc. Vicente Enrique	Escuela Profesional de	Compiladores
Machaca Arceda	Ingeniería de Software	

PRÁCTICA	TEMA	DURACIÓN
01	Introducción	3 horas

Integrantes:

Carlos Corrales

Leonardo Deza

1. Ejercicios

1.1 Redacta el siguiente código, genera el código ensamblador y explica en qué parte (del código ensamblador) se definen las variables c y m.

```
int main(){
char* c = "abcdef";
int m = 11148;

return 0;
}
```

```
.file "1.cpp"
        .text
        .section .rdata, "dr"
_ZStL19piecewise_construct:
       .space 1
.lcomm _ZStL8__ioinit,1,1
       .def __main; .scl 2;
                                      .type 32;
                                                     .endef
.LC0:
       .ascii "abcdef\0"
       .text
       .globl main
       .def main; .scl 2;
                                     .type 32;
                                                     .endef
       .seh_proc
                      main
main:
.LFB1573:
       pushq %rbp
       .seh_pushreg
                     %rbp
       pushq %rbx
       .seh_pushreg
                      %rbx
       subq $88, %rsp
       .seh_stackalloc 88
       leaq 128(%rsp), %rbp
       .seh_setframe %rbp, 128
       .seh_endprologue
       call __main
               -53(%rbp), %rax
       lead
             %rax, %rcx
       movq
               _ZNSaIcEC1Ev
       call
             -53(%rbp), %rdx
       leaq
              -96(%rbp), %rax
       leaq
       movq
              %rdx, %r8
              .LCO(%rip), %rdx
       lead
       movq %rax, %rcx
.LEHB0:
              _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEEC1EPKcRKS3_
       call
.LEHE0:
              -53(%rbp), %rax
       leaq
       movq
              %rax, %rcx
                _ZNSaIcED1Ev
       call
       movl
               $11148, -52(%rbp)
             $0, %ebx
       mov1
       leaq
               -96(%rbp), %rax
              %rax, %rcx
       mova
       call
               _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEED1Ev
       movl %ebx, %eax
       jmp
               .L5
.L4:
             %rax, %rbx
       movq
              -53(%rbp), %rax
       leaq
              %rax, %rcx
       movq
               ZNSaIcED1Ev
       call
              %rbx, %rax
       movq
              %rax, %rcx
       movq
.LEHB1:
       call _Unwind_Resume
.LEHE1:
.L5:
              $88, %rsp
       addq
       popq
              %rbx
              %rbp
       popq
       ret
                 _gxx_personality_seh0; .scl 2;
       .def
                                                                    .endef
                                                     .type 32;
       .seh_handler __gxx_personality_seh0, @unwind, @except .seh_handlerdata
.LLSDA1573:
       .byte 0xff
       .byte 0xff
.byte 0x1
        .uleb128 .LLSDACSE1573-.LLSDACSB1573
```

```
.LLSDACSB1573:
            .uleb128 .LEHB0-.LFB1573
.uleb128 .LEHE0-.LEHB0
            .uleb128 .L4-.LFB1573 .uleb128 0
            .uleb128 .LEHB1-.LFB1573
.uleb128 .LEHE1-.LEHB1
            .uleb128 0
              .uleb128 0
.LLSDACSE1573:
  .def __tcf_0; .scl 3; .type 32; .endef __tcf_0: __tcf_0
           .text
.LFB2058:
            pushq %rbp
             .seh_pushreg %rb
movq %rsp, %rbp
                                     %rbp
             movq
            .seh_setframe %rt
subq $32, %rsp
.seh_stackalloc 32
                                    %rbp, 0
            .seh_endprologue
leaq _ZStl8__ioinit(%rip), %rcx
call _ZNSt8ios_base4InitD1Ev
             nop
            addq $32, %rsp
popq %rbp
             .def _Z41_static_initialization_and_destruction_0ii;
.seh_proc _Z41_static_initialization_and_destruction_0ii;
        _____selic_initialization_and_destruction_0ii; .scl 3; .type 32;
_seh_proc __Z41__static_initialization_and_destruction_0ii
_static_initialization_and_destruction_0ii:
057:
             .seh_endproc
            pushq %rbp
             .seh_pushreg
            movq %rsp, %rbp
.seh_setframe %rb
                                      %rbp, 0
            subq $32, %rsp
.seh_stackalloc 32
              .seh_endprologue
                         %ecx, 16(%rbp)
%edx, 24(%rbp)
             mov1
             mov1
                      $1, 16(%rbp)
             cmp1
             jne
             cmpl $65535, 24(%rbp)
            leaq _ZStL8_ioinit(%rip), %rcx
call _ZNSt8ios_base4InitC1Ev
leaq _tcf-0(%rip), %rcx
call atexit
                        .L9
.L9:
             nop
            addq $32, %rsp
popq %rbp
             ret
            .seh_endproc
            .def _GLOBAL_sub_I_main; .scl 3; .type 32; .endef .seh_proc _GLOBAL_sub_I_main
_uLOBAL__sub_I_main:
.LFB2059:
           pushq %rbp
              .seh_pushreg
                                      %rbp
             movq
                         %rsp, %rbp
            .seh_setframe %rbp, 0
subq $32, %rsp
.seh_stackalloc 32
             .seh_endprologue
mov1 $65535,
            movl $65535, %edx
movl $1, %ecx
call Z41_static_initialization_and_destruction_0ii
            nop
addq $32, %rsp
nopq %rbp
             .seh_endproc
                                      .ctors."w"
             .section
             .align 8
            _GLOBAL__sub_I_main
.ident "GCC: (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0"
                       "GCC: (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0"

_ZNSAICECIEv; .scl 2; .type 32; .endef
_ZNST__cxx1112basic_stringIcSt11char_traitsIcESaICEECIEPKcRKS3_;

_ZNSAICEDIEv; .scl 2; .type 32; .endef
_ZNSTT__cxx1112basic_stringIcSt11char_traitsIcESaICEEDIEv; .sc
_Unwind_Resume; .scl 2; .type 32; .endef
_ZNSTBios_base4InitDIEv; .scl 2; .type 32; .en
_ZNSTBios_base4InitCIEv; .scl 2; .type 32; .en
atexit; .scl 2; .type 32; .endef
             .def
                                                                                                                                            .scl 2; .type 32; .endef
             .def
             .def
                                                                                                                             .scl 2; .type 32; .endef
             .def
             .def
                                                                                                                                 .endef
                                                                                                                                 .endef
             .def
             .def
```

```
.ascii "abcdef\0" movl $11148, -52(%rbp)
```

1.2 Redacta el siguiente código, genera el código ensamblador y explica en qué parte (del código ensamblador) se define la división entre 8.

```
int main() {
  char* c = "abcdef";
  int m = 11148;
  int x = m/8;

return 0;
}
```

```
.LLSDA1573: -
          .byte 0xff
.byte 0xff
.byte 0x1
            .uleb128 .LLSDACSE1573-.LLSDACSB1573
.LLSDACSB1573:
           .uleb128 .LEHB0-.LFB1573
           .uleb128 .LEHE0-.LEHB0 .uleb128 .L4-.LFB1573
           .uleb128 0
.uleb128 .LEHB1-.LFB1573
.uleb128 .LEHE1-.LEHB1
            .uleb128 0
            .uleb128 0
.LLSDACSE1573:
           .text
            .seh_endproc
           .def __tcf_0; .scl 3;
.seh_proc __tcf_0
                                                                                         .endef
                                                                .type 32;
   tcf 0:
.LFB2058:
           pushq %rbp
            .seh_pushreg
                                  %rbp
                      %rsp, %rbp
            movq
            .seh_setframe %r
subq $32, %rsp
                                 %rbp, 0
           subq $32, %rsp
.seh_stackalloc 32
            .seh_endprologue
leaq _ZStl8__ioinit(%rip), %rcx
call _ZNSt8ios_base4InitD1Ev
            nop
addq $32, %rsp
popq %rbp
            ret
            .seh_endproc
         .def Z41_static_initialization_and_destruction_0ii; .scl 3; .seh_proc _Z41_static_initialization_and_destruction_0ii static_initialization_and_destruction_0ii:
                                                                                                                              .type 32; .endef
.LFB2057:
           pushq %rbp
            .seh_pushreg
                                  %rbp
                      %rsp, %rbp
            mova
            .seh_setframe %r
subq $32, %rsp
                                 %rbp, 0
           subq $32, %rsp
.seh_stackalloc 32
            .seh_endprologue
                       %ecx, 16(%rbp)
%edx, 24(%rbp)
            mov1
            mov1
           cmpl $1, 16(%rbp)
jne .L9
            cmpl $65535, 24(%rbp)
            ine
                      .L9
            leaq
                      _ZStL8__ioinit(%rip), %rcx
                   _ZNSt8ios_base4InitC1Ev
__tcf_0(%rip), %rcx
atexit
            call
            leaq
            call
.L9:
            nop
                     $32, %rsp
%rbp
            addq
            popq
            .seh_endproc
            .def
                    _GLOBAL__sub_I_main; .scl 3;
                                                                                .type 32;
                                                                                                     .endef
                               _GLOBAL__sub_I_main
            .seh_proc
            _sub_I_main:
 GLOBAL
 .LFB2059:
           pushq %rbp
            .seh_pushreg
            movq
                      %rsp, %rbp
            .seh_setframe
                                  %rbp. 0
                      $32, %rsp
            subq
            .seh_stackalloc 32
            .seh_endprologue
           mov1 $65535, %edx
mov1 $1, %ecx
call _241_static_initialization_and_destruction_0ii
            nop
            addq $32, %rsp
            popq
            ret
            .seh_endproc
            .section
                                  .ctors, "w"
            .align 8
            .quad _GLOBAL__sub_I_main
.ident "GCC: (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0"
.def _ZNSaIcEC1Ev; .scl 2; .type 32; .endef
                      "GCC: (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0"
_ZNSaIcECIEv; .scl 2; .type 32; .endef
_ZNSt7_cxx1112basic_stringIcSt11char_traitsIcESaIcEECIEPKcRKS3;
_ZNSaIcED1Ev; .scl 2; .type 32; .endef
_ZNSt7_cxx1112basic_stringIcSt11char_traitsIcESaIcEED1Ev; .scl
_Unwind_Resume; .scl 2; .type 32; .endef
_ZNStBios_base4InitD1Ev; .scl 2; .type 32; .ender
_ZNStBios_base4InitC1Ev; .scl 2; .type 32; .ender
atexit; .scl 2; .type 32; .endef
            .def
                                                                                                                            .scl 2; .type 32;
                                                                                                                                                                              .endef
            .def
                                                                                                                 .scl 2; .type 32;
            .def
            .def
                                                                                                                .endef
            .def
            .def
                                                                                                                   .endef
            .def
```

```
.file "2.cpp"
       .text
       .section .rdata, "dr"
_ZStL19piecewise_construct:
       .space 1
.lcomm _ZStL8__ioinit,1,1
             __main; .scl 2;
                                     .type 32;
                                                     .endef
       .def
.LC0:
       .ascii "abcdef\0"
       .text
       .globl main
       .def main; .scl 2;
                                    .type 32;
                                                     .endef
       .seh_proc
                     main
main:
.LFB1573:
       pushq %rbp
       .seh_pushreg
                     %rbp
       pushq %rbx
                     %rbx
       .seh_pushreg
       subq $88, %rsp
       .seh_stackalloc 88
       leaq 128(%rsp), %rbp
       .seh_setframe %rbp, 128
       .seh_endprologue
             __main
       call
       leaq
               -57(%rbp), %rax
       movq
              %rax, %rcx
               _ZNSaIcEC1Ev
       call
       leaq
              -57(%rbp), %rdx
              -96(%rbp), %rax
       leaq
       movq
              %rdx, %r8
       leaq
              .LCO(%rip), %rdx
       movq
             %rax, %rcx
.LEHB0:
             _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEEC1EPKcRKS3_
       call
.LEHE0:
              -57(%rbp), %rax
       leaq
       movq
              %rax, %rcx
               _ZNSaIcED1Ev
       call
       mov1
              $11148, -52(%rbp)
              -52(%rbp), %eax
       movl
       leal
              7(%rax), %edx
              %eax, %eax
       testl
       cmovs
              %edx, %eax
       sarl
               $3, %eax
              %eax, -56(%rbp)
       movl
       movl
              $0, %ebx
       leaq
               -96(%rbp), %rax
              %rax, %rcx
       mova
       call
               _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEED1Ev
       mov1
              %ebx, %eax
       jmp
               .L5
.L4:
       movq
              %rax, %rbx
               -57(%rbp), %rax
       leaq
       movq
               %rax, %rcx
               _ZNSaIcED1Ev
       call
       movq
              %rbx, %rax
       movq
              %rax, %rcx
.LEHB1:
       call
              _Unwind_Resume
.LEHE1:
.L5:
       addq
               $88, %rsp
       popq
               %rbx
       popq
               %rbp
       ret
       .def
                _gxx_personality_seh0; .scl 2;
                                                    .type 32;
                                                                    .endef
       .seh_handler __gxx_personality_seh0, @unwind, @except
       .seh_handlerdata
```

```
movl -52(%rbp), %eax
leal 7(%rax), %edx
testl %eax, %eax
cmovs %edx, %eax
sarl $3, %eax
movl %eax, -56(%rbp)
```

1.3 Redacta el siguiente código, genera el código ensamblador y explica en qué parte (del código ensamblador) se define la división entre 4.

```
int main() {
  char* c = "abcdef";
  int m = 11148;
  int x = m/8;
  int y = m/4;
  int z = m/2;
  return 0;
}
```

```
.file "3.cpp"
        .text
        .section .rdata,"dr"
_ZStL19piecewise_construct:
        .space 1
.lcomm _ZStL8__ioinit,1,1
               __main; .scl
        .def
                                2;
                                        .type 32;
                                                         .endef
.LC0:
        .ascii "abcdef\0"
        .text
        .globl main
        .def main;
                       .scl
                                2;
                                        .type 32;
                                                         .endef
        .seh_proc
                        main
main:
.LFB1573:
       pushq %rbp
        .seh_pushreg
                        %rbp
        pushq %rbx
        .seh_pushreg
                        %rbx
               $104, %rsp
        subq
        .seh_stackalloc 104
        leaq 128(%rsp), %rbp
        .seh_setframe %rbp, 128
        .seh_endprologue
                __main
-49(%rbp), %rax
        call
        leaq
        movq
                %rax, %rcx
        call
                _ZNSaIcEC1Ev
                -49(%rbp), %rdx
        leaq
                -96(%rbp), %rax
        leaq
        movq
                %rdx, %r8
                .LC0(%rip), %rdx
        leag
        movq
                %rax, %rcx
.LEHB0:
        call
                _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEEC1EPKcRKS3_
.LEHE0:
                -49(%rbp), %rax
        leaq
                %rax, %rcx
        movq
        call
                _ZNSaIcED1Ev
        mov1
                $11148, -36(%rbp)
                -36(%rbp), %eax
        mov1
        leal
                7(%rax), %edx
        testl
                %eax, %eax
        cmovs
                %edx, %eax
        sarl
                $3, %eax
                %eax, -40(%rbp)
        mov1
        movl
                -36(%rbp), %eax
                3(%rax), %edx
        leal
        testl
                %eax, %eax
        cmovs
                %edx, %eax
                $2, %eax
        sarl
                %eax, -44(%rbp)
        movl
                -36(%rbp), %eax
        movl
                %eax, %edx
        mov1
        shrl
                $31, %edx
                %edx, %eax
        addl
        sarl
                %eax
        movl
                %eax, -48(%rbp)
        mov1
                $0, %ebx
        leaq
                -96(%rbp), %rax
        movq
                %rax, %rcx
                _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEED1Ev %ebx, %eax
        call
        movl
        jmp
                .L5
.L4:
        movq
                %rax, %rbx
                -49(%rbp), %rax
        lead
        movq
                %rax, %rcx
                _ZNSaIcED1Ev
%rbx, %rax
        call
        movq
        movq
                %rax, %rcx
.LEHB1:
       call
                _Unwind_Resume
```

```
.LEHE1:
.15:
         addq
                $104, %rsp
         popq
                  %rbx
         popq
                %rbp
         ret
         .... __gxx_personality_seh0; .scl 2; .type 32; .seh_handler __gxx_personality_seh0, @unwind, @except .seh_handlerdata
        .def
                                                                                 .endef
.LLSDA1573:
         .byte
                 0xff
         .byte 0xff
         .byte 0x1
         .uleb128 .LLSDACSE1573-.LLSDACSB1573
.LLSDACSB1573:
        .uleb128 .LEHB0-.LFB1573
        .uleb128 .LEHE0-.LEHB0
         .uleb128 .L4-.LFB1573
         .uleb128 0
        .uleb128 .LEHB1-.LFB1573
         .uleb128 .LEHE1-.LEHB1
        .uleb128 0
         .uleb128 0
.LLSDACSE1573:
 .sen_enaproc
.def __tcf_0; .scl 3; .type 32; .endef
.seh_proc __tcf_0
_tcf_0:
.LFB2058:
        pushq %rbp
        .seh_pushreg
        movq %rsp, %rbp
        .seh_setframe %rbp, 0
         subq $32, %rsp
         .seh_stackalloc 32
         .seh_endprologue
        leaq _ZStL8__ioinit(%rip), %rcx
call _ZNSt8ios_base4InitD1Ev
         nop
        addq $32, %rsp
popq %rbp
         ret
        .seh_endproc
         .def _Z41__static_initialization_and_destruction_0ii;
.seh_proc _Z41__static_initialization_and_destruction_0ii
                                                                                 .scl 3; .type 32; .endef
_Z41__static_initialization_and_destruction_0ii:
.LFB2057:
        pushq %rbp
         .seh_pushreg
                          %rbp
         movq %rsp, %rbp
         .seh_setframe %rbp, 0
         subq $32, %rsp
        .seh stackalloc 32
         .seh_endprologue
         movl %ecx, 16(%rbp)
        movl %edx, 24(%rbp)
cmpl $1, 16(%rbp)
jne .L9
         cmpl $65535, 24(%rbp)
                 .L9
         jne
        leaq _ZStL8_ioinit(%rip), %rcx
call _ZNSt8ios_base4InitC1Ev
leaq _tcf_0(%rip), %rcx
call atexit
```

```
.L9:
         addq
                  $32, %rsp
         popq
                  %rbp
         ret
         .seh_endproc
                _GLOBAL__sub_I_main;
                                             .scl
                                                                 .type 32;
                                                                                    .endef
         .seh proc
                           _GLOBAL__sub_I_main
_GLOBAL__sub_I_main:
.LFB2059:
         pushq
         .seh_pushreg
                           %rbp
                 %rsp, %rbp
etframe %rbp, 0
         movq
         .seh_setframe
                 $32, %rsp
         subq
         .seh_stackalloc 32
         .seh_endprologue
                 $65535, %edx
         movl
         mov1
                 $1, %ecx
_Z41__static_initialization_and_destruction_0ii
         call
                  $32, %rsp
         adda
         popq
                  %rbp
         ret
         .seh_endproc
         .section
                            .ctors,"w"
         .align 8
         .quad _GLOBAL__sub_I_main
.ident "GCC: (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0"
                  _ZNSaIcEC1Ev; .scl 2;
                                                      .type 32;
                                                                          .endef
                   _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEEC1EPKcRKS3_;
                                                                                                       .scl
                                                                                                                         .type
                                                                                                                                            .endef
                 ZNSalcED1Ev; scl 2; type 32; endef
ZNST_cxx1112basic_stringIcSt11char_traitsIcESalcEED1Ev;
Unwind_Resume; scl 2; type 32; endef
ZNSt8ios_base4InitD1Ev; scl 2; type 32
Type 32
         .def
                                                                                             .scl
                                                                                                      2:
                                                                                                                .type 32;
                                                                                                                                   .endef
         .def
         .def
                                                                                             .endef
         .def
                   ZNSt8ios_base4InitC1Ev;
                                                        .scl
                                                                          .type
                                                                                             .endef
                                                                 .endef
                  atexit; .scl 2;
                                           .type 32;
         .def
```

```
movl -36(%rbp), %eax
leal 3(%rax), %edx
testl %eax, %eax
cmovs %edx, %eax
sarl $2, %eax
movl %eax, -44(%rbp)
```

1.4 Redacta el siguiente código, genera el código ensamblador y explica en qué parte (del códigoensamblador) se define la división entre 2.

```
int main() {
  char* c = "abcdef";
  int m = 11148;
  int x = m/8;
  int y = m/4;
  int z = m/2;
  return 0;
}
```

```
.file "3.cpp"
        .text
        .section .rdata, "dr"
_ZStL19piecewise_construct:
       .space 1
.lcomm _ZStL8__ioinit,1,1
              __main; .scl
                              2;
                                       .type 32;
                                                       .endef
.LC0:
       .ascii "abcdef\0"
       .text
       .globl main
       .def main;
                      .scl
                               2;
                                       .type 32;
                                                       .endef
       .seh_proc
                       main
main:
.LFB1573:
       pushq %rbp
       .seh_pushreg
                       %rbp
       pushq %rbx
       .seh_pushreg
                       %rbx
              $104, %rsp
       subq
        .seh_stackalloc 104
       leaq 128(%rsp), %rbp
       .seh_setframe %rbp, 128
        .seh_endprologue
               __main
       call
               -49(%rbp), %rax
       leaq
       movq
              %rax, %rcx
               _ZNSaIcEC1Ev
       call
               -49(%rbp), %rdx
       leaq
               -96(%rbp), %rax
       leaq
       movq
               %rdx, %r8
               .LCO(%rip), %rdx
       leag
       movq
              %rax, %rcx
.LEHB0:
       call
               _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEEC1EPKcRKS3_
.LEHE0:
               -49(%rbp), %rax
       leaq
               %rax, %rcx
       movq
       call
               _ZNSaIcED1Ev
       mov1
               $11148, -36(%rbp)
               -36(%rbp), %eax
       movl
       leal
               7(%rax), %edx
               %eax, %eax
%edx, %eax
       testl
       CMOVS
               $3, %eax
       sarl
               %eax, -40(%rbp)
       movl
       movl
               -36(%rbp), %eax
               3(%rax), %edx
       leal
       testl %eax, %eax
       cmovs
               %edx, %eax
               $2, %eax
       sarl
               %eax, -44(%rbp)
       movl
               -36(%rbp), %eax
       mov1
               %eax, %edx
       mov1
        shrl
               $31, %edx
       addl
               %edx, %eax
       sarl
               %eax
       movl
               %eax, -48(%rbp)
       mov1
               $0, %ebx
               -96(%rbp), %rax
       leaq
               %rax, %rcx
       movq
               _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEED1Ev
       call
               %ebx, %eax
       movl
       jmp
               .L5
.L4:
       movq
               %rax, %rbx
               -49(%rbp), %rax
       leaq
       movq
               %rax, %rcx
       call
                ZNSaIcED1Ev
               %rbx, %rax
       movq
       movq
               %rax, %rcx
.LEHB1:
       call
               _Unwind_Resume
```

```
.LEHE1:
.15:
         addq
                $104, %rsp
         popq
                  %rbx
         popq
                %rbp
         ret
         .... __gxx_personality_seh0; .scl 2; .type 32; .seh_handler __gxx_personality_seh0, @unwind, @except .seh_handlerdata
        .def
                                                                                 .endef
.LLSDA1573:
         .byte
                 0xff
         .byte 0xff
         .byte 0x1
         .uleb128 .LLSDACSE1573-.LLSDACSB1573
.LLSDACSB1573:
        .uleb128 .LEHB0-.LFB1573
        .uleb128 .LEHE0-.LEHB0
         .uleb128 .L4-.LFB1573
         .uleb128 0
        .uleb128 .LEHB1-.LFB1573
         .uleb128 .LEHE1-.LEHB1
        .uleb128 0
         .uleb128 0
.LLSDACSE1573:
 .sen_enaproc
.def __tcf_0; .scl 3; .type 32; .endef
.seh_proc __tcf_0
_tcf_0:
.LFB2058:
        pushq %rbp
        .seh_pushreg
        movq %rsp, %rbp
        .seh_setframe %rbp, 0
         subq $32, %rsp
         .seh_stackalloc 32
         .seh_endprologue
        leaq _ZStL8__ioinit(%rip), %rcx
call _ZNSt8ios_base4InitD1Ev
         nop
        addq $32, %rsp
popq %rbp
         ret
        .seh_endproc
         .def _Z41__static_initialization_and_destruction_0ii;
.seh_proc _Z41__static_initialization_and_destruction_0ii
                                                                                 .scl 3; .type 32; .endef
_Z41__static_initialization_and_destruction_0ii:
.LFB2057:
        pushq %rbp
         .seh_pushreg
                          %rbp
         movq %rsp, %rbp
         .seh_setframe %rbp, 0
         subq $32, %rsp
        .seh stackalloc 32
         .seh_endprologue
         movl %ecx, 16(%rbp)
        movl %edx, 24(%rbp)
cmpl $1, 16(%rbp)
jne .L9
         cmpl $65535, 24(%rbp)
                 .L9
         jne
        leaq _ZStL8_ioinit(%rip), %rcx
call _ZNSt8ios_base4InitC1Ev
leaq _tcf_0(%rip), %rcx
call atexit
```

```
.L9:
        adda
                 $32, %rsp
        popq
                %rbp
        ret
        .seh_endproc
        .def
                _GLOBAL__sub_I_main;
                                           .scl
                                                            .type
                                                                  32;
                                                                             .endef
        .seh proc
                         _GLOBAL__sub_I_main
GLOBAL
        __sub_I_main:
.LFB2059:
        pushq
        .seh_pushreg
                         %rbp
                %rsp, %rbp
tframe %rbp, 0
        movq
        .seh_setframe
                $32, %rsp
        subq
        .seh_stackalloc 32
        .seh_endprologue
                $65535, %edx
        movl
        mov1
                $1, %ecx
        call
                _Z41__static_initialization_and_destruction_0ii
        nop
                 $32, %rsp
        adda
        popq
                %rbp
        ret
        .seh_endproc
        .section
                          .ctors,"w"
         .align 8
                 _GLOBAL__sub_I_main

"GCC: (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0"
        . auad
         .ident
        .def
                 _ZNSaIcEC1Ev;
                                 .scl
                                                   .type 32;
                                                                    .endef
         .def
                 _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEEC1EPKcRKS3_;
                                                                                              .scl
                                                                                                                .type
                                                                                                                                 .endef
         .def
                _ZNSaICED1Ev; .scl 2; .type 32; .endef
_ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEED1Ev;
        .def
                                                                                      .scl
                                                                                              2:
                                                                                                       .type
                                                                                                               32:
                                                                                                                        .endef
                 _Unwind_Resume; .scl
         .def
                                         2;
                                                   .type
                                                           32;
                                                                    .endef
                _ZNSt8ios_base4InitD1Ev;
                                                   .scl
                                                            2;
                                                                                      .endef
                                                                     .type
         .def
                  ZNSt8ios_base4InitC1Ev;
                                                   .scl
                                                                    .type
                                                                             32;
                                                                                      .endef
                                                            .endef
                                           .type 32;
         .def
                 atexit; .scl
                                2;
```

```
movl -36(%rbp), %eax
movl %eax, %edx
shrl $31, %edx
addl %edx, %eax
sarl %eax
movl %eax, -48(%rbp)
```

1.5 Redacta el siguiente código, genera el código ensamblador y explica:

En qué parte del código ensamblador se define la función div4.

En qué parte del código ensamblador se invoca a la función div4.

En qué parte del código ensamblador dentro de la función div4 se procesa la división.

```
int div4(int x){
return x/4;
}

int main(){
  char* c = "abcdef";
  int m = 11148;
  int x = m/8;
  int y = m/4;
  int z = m/2;

int rpt = div4(5);

return 0;
}
```

```
.file "5.cpp"
        .text
        .section .rdata,"dr"
_ZStL19piecewise_construct:
        .space 1
.lcomm _ZStL8__ioinit,1,1
        .text
       .globl _Z4div4i
.def _Z4div4i;
                               .scl
                                     2;
                                               .type 32;
                                                               .endef
                     _Z4div4i
        .seh_proc
_Z4div4i:
.LFB1573:
       pushq %rbp
        .seh_pushreg
                       %rbp
        movq %rsp, %rbp
        .seh_setframe %rbp, 0
        .seh_endprologue
        movl %ecx, 16(%rbp)
        movl
               16(%rbp), %eax
             3(%rax), %edx
       leal
        testl %eax, %eax
        cmovs
               %edx, %eax
               $2, %eax
        sarl
               %rbp
        popq
        ret
        .seh_endproc
        .def
              __main; .scl
                               2;
                                       .type 32;
                                                       .endef
        .section .rdata, "dr"
.LC0:
        .ascii "abcdef\0"
        .text
        .globl main
        .def main;
                       .scl
                               2;
                                       .type 32;
                                                        .endef
        .seh_proc
                       main
main:
.LFB1574:
       pushq %rbp
        .seh_pushreg
                       %rbp
        pushq %rbx
        .seh_pushreg
                       %rbx
        subq $104, %rsp
        .seh_stackalloc 104
       leaq 128(%rsp), %rbp
.seh_setframe %rbp, 128
        .seh_endprologue
              __main
-53(%rbp), %rax
        call
        leaq
               %rax, %rcx
        mova
               _ZNSaIcEC1Ev
        call
        leaq
                -53(%rbp), %rdx
        leag
               -96(%rbp), %rax
       movq
               %rdx, %r8
       leaq
                .LC0(%rip), %rdx
               %rax, %rcx
       movq
.LEHB0:
        call
               _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEEC1EPKcRKS3_
.LEHE0:
       leag
                -53(%rbp), %rax
               %rax, %rcx
        mova
        call
                _ZNSaIcED1Ev
        mov1
                $11148, -36(%rbp)
               -36(%rbp), %eax
        mov1
        leal
                7(%rax), %edx
        testl
                %eax, %eax
               %edx, %eax
       cmovs
        sarl
                $3, %eax
        movl
                %eax, -40(%rbp)
                -36(%rbp), %eax
        movl
       leal
                3(%rax), %edx
        testl
               %eax, %eax
        cmovs
               %edx, %eax
        sarl
                $2, %eax
               %eax, -44(%rbp)
       movl
                -36(%rbp), %eax
        movl
       mov1
               %eax, %edx
```

```
movl
                %eax, %edx
        shrl
                 $31, %edx
        addl
                 %edx, %eax
        sarl
                 %eax
        movl
                 %eax, -48(%rbp)
                $5, %ecx
        mov1
                 _Z4div4i
%eax, -52(%rbp)
        call
        mov1
                 $0, %ebx
        movl
                 -96(%rbp), %rax
        leaq
        movq
                 %rax, %rcx
                _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEED1Ev %ebx, %eax
        call
        movl
        jmp
.L6:
                 %rax, %rbx
        movq
                 -53(%rbp), %rax
        lead
        movq
                %rax, %rcx
        call
                  _ZNSaIcED1Ev
                %rbx, %rax
%rax, %rcx
        movq
        movq
.LEHB1:
        call
                _Unwind_Resume
.LEHE1:
.L7:
        addq
               $104, %rsp
        popq
                 %rbx
                %rbp
        popq
        ret
        .seh_handler _gxx_personality_seh0; .scl 2; .type 3
.seh_handler _gxx_personality_seh0, @unwind, @except
.seh_handlerdata
                                                             .type 32;
                                                                                .endef
.LLSDA1574:
         .byte 0xff
         .byte 0xff
         .byte 0x1
         .uleb128 .LLSDACSE1574-.LLSDACSB1574
.LLSDACSB1574:
        .uleb128 .LEHB0-.LFB1574
         .uleb128 .LEHE0-.LEHB0
        .uleb128 .L6-.LFB1574
         .uleb128 0
        .uleb128 .LEHB1-.LFB1574
         .uleb128 .LEHE1-.LEHB1
        .uleb128 0
         .uleb128 0
.LLSDACSE1574:
        .text
         .seh_endproc
        .def __tcf_0; .scl 3; .type 32; .seh_proc __tcf_0
                                                                    .endef
 _tcf_0:
.LFB2059:
       pushq %rbp
         .seh_pushreg
                          %rbp
        movq %rsp, %rbp
         .seh_setframe %rbp, 0
        subq $32, %rsp
         .seh_stackalloc 32
         .seh_endprologue
        leaq _ZStL8_ioinit(%rip), %rcx
call _ZNSt8ios_base4InitD1Ev
        nop
        addq $32, %rsp
popq %rbp
        ret
         .seh_endproc
.def _Z41_static_initialization_and_destruction_0ii; .scl 3; .type 32; .seh_proc _Z41_static_initialization_and_destruction_0ii _Z41_static_initialization_and_destruction_0ii:
                                                                                                                   .endef
.LFB2058:
        pushq %rbp
         .seh_pushreg
                          %rbp
        movq %rsp, %rbp
```

```
.seh_setframe %rbp, 0
subq $32, %rsp
          subq $32, %rsp
.seh_stackalloc 32
           .seh_endprologue
                     %ecx, 16(%rbp)
           movl
                     %edx, 24(%rbp)
           mov1
                     $1, 16(%rbp)
           cmpl
           ine
                     .L11
                     $65535, 24(%rbp)
           cmpl
           jne
           leaq
                     _ZStL8__ioinit(%rip), %rcx
                     _ZNSt8ios_base4InitC1Ev
__tcf_0(%rip), %rcx
atexit
           call
           leaq
           call
.L11:
           nop
           addq
                     $32, %rsp
           popq
                     %rbp
           ret
ret
.seh_endproc
.def _GLOBAL__sub_I__Z4div4i;
.seh_proc _GLOBAL__sub_I__Z4div4i
_GLOBAL__sub_I__Z4div4i:
                                                                .scl 3;
                                                                                      .type 32;
                                                                                                             .endef
.LFB2060:
          pushq %rbp
           .seh_pushreg
                                %rbp
                    %rsp, %rbp
           movq
           .seh_setframe
                               %rbp, 0
           subq $32, %rsp
           .seh_stackalloc 32
           .seh_endprologue
                   $65535, %edx
           movl
                    $1, %ecx
_Z41__static_initialization_and_destruction_0ii
           movl
           call
                     $32, %rsp
           addq
          popq
ret
                     %rbp
           .seh_endproc
           .section
                                .ctors, "w"
           .align 8
           .duad _GLOBAL__sub_I__Z4div4i
.ident "GCC: (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 8.1.0"
.def _ZNSaICEC1Ev; .scl 2; .type 32; .endef
.def _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaICEEC1EPKCRKS3_;
.def _ZNSaICED1Ev; .scl 2; .type 32; .endef
                                                                                                                        .scl
                                                                                                                                             .type 32;
                                                                                                                                                                    .endef
                                                                                                                                  2;
           .def
                     _ZNSt7__cxx1112basic_stringIcSt11char_traitsIcESaIcEED1Ev;
                                                                                                             .scl
                                                                                                                        2;
                                                                                                                                   .type 32;
                     Unwind_Resume; .scl 2; .type 32; .endef
_ZNSt8ios_base4InitD1Ev; .scl 2; .type
_ZNSt8ios_base4InitC1Ev; .scl 2; .type
           .def
           .def
                                                                                       .type 32;
                                                                                                             .endef
                     __ZNSt8ios_base4InitC1Ev;
atexit; .scl 2; .
           .def
                                                                                       .type 32;
                                                                                                             .endef
           .def
                                                      .type 32;
                                                                            .endef
```

.LFB1573:

pushq %rbp

```
1.-
       lcomm ZStL8_ioinit,1,1
       .text
       .globl _Z4div4i
               _Z4div4i;
                                             .type 32;
       .def
                              .scl
                                     2;
                                                             .endef
                      _Z4div4i
       .seh_proc
2.-
       call
               Z4div4i
3.-
```

```
.seh_pushreg
               %rbp
movq %rsp, %rbp
.seh_setframe %rbp, 0
. seh\_end prologue
movl
       %ecx, 16(%rbp)
       16(%rbp), %eax
movl
leal
       3(%rax), %edx
testl
       %eax, %eax
cmovs %edx, %eax
sarl
       $2, %eax
       %rbp
popq
ret
.seh_endproc
.def
       __main;
                      .scl
                              2;
                                             32;
                                                     .endef
                                     .type
.section .rdata,"dr"
```

1.6 Redacta el siguiente código, genera el código ensamblador y explica:

En qué parte del código ensamblador se define la función div.

En qué parte del código ensamblador se invoca a la función div.

En qué parte del código ensamblador dentro de la función div se procesa la división.

```
int div(int x, int y){
return x/y;
}

int div4(int x){
return x/4;
}

int main(){
  char* c = "abcdef";
  int m = 11148;
  int x = m/8;
  int y = m/4;
  int z = m/2;

int rpt = div(5,4);
  int rpt2 = div4(5);

return 0;
}
```

```
.file "Ejercicio6.cpp"
       .text
       .globl __Z3divii
.def __Z3divii; .scl 2; .type 32;
                                                          .endef
 Z3divii:
LFB0:
       .cfi_startproc
       push1 %ebp
       .cfi_def_cfa_offset 8
       .cfi_offset 5, -8
       movl %esp, %ebp
       .cfi_def_cfa_register 5
       movl 8(%ebp), %eax
       cltd
       idivl 12(%ebp)
popl %ebp
       .cfi_restore 5
       .cfi_def_cfa 4, 4
       ret
       .cfi_endproc
LFE0:
       .globl __Z4div4i
.def __Z4div4i; .scl 2; .type 32;
                                                          .endef
 Z4div4i:
LFB1:
       .cfi_startproc
       push1 %ebp
       .cfi_def_cfa_offset 8
       .cfi_offset 5, -8
       movl %esp, %ebp
       .cfi_def_cfa_register 5
       movl 8(%ebp), %eax
       cltd
       and1
              $3, %edx
              %edx, %eax
       addl
            $2, %eax
       sarl
       popl
            %ebp
       .cfi_restore 5
       .cfi_def_cfa 4, 4
       ret
       .cfi_endproc
LFE1:
                            .scl 2; .type 32; .endef
       .def ___main;
       .section .rdata, "dr"
LC0:
       .ascii "abcdef\0"
       .text
       .globl _main
       .def _main; .scl 2; .type 32;
                                                   .endef
_main:
```

```
LFB2:
                .cfi_startproc
               push1 %ebp
                .cfi_def_cfa_offset 8
                .cfi_offset 5, -8
movl %esp, %ebp
               mov1
                .cfi_def_cfa_register 5
                andl
                       $-16, %esp
                       $48, %esp
               subl
                         main
               call
                       $LC0, 44(%esp)
               movl
                       $11148, 40(%esp)
               movl
               movl
                       40(%esp), %eax
                cltd
                       $7, %edx
                andl
                addl
                       %edx, %eax
                sarl
                       $3, %eax
                       %eax, 36(%esp)
                movl
                       40(%esp), %eax
               movl
                cltd
                andl
                       $3, %edx
                addl
                       %edx, %eax
                       $2, %eax
                sarl
                movl
                       %eax, 32(%esp)
                       40(%esp), %eax
                movl
                movl
                       %eax, %edx
                       $31, %edx
%edx, %eax
                shrl
                addl
                       %eax
                sarl
                       %eax, 28(%esp)
               movl
               movl
                       $4, 4(%esp)
                       $5, (%esp)
               movl
                        __Z3divii
                call
                       movl
               mov1
               call
                        _Z4div4i
               movl
                       %eax, 20(%esp)
               mov1
                       $0, %eax
               leave
                .cfi_restore 5
                .cfi_def_cfa 4, 4
               ret
                .cfi_endproc
        LFE2:
                .ident "GCC: (MinGW.org GCC-6.3.0-1) 6.3.0"
.globl __Z3divii
.def __Z3divii; .scl 2; .type 32; .endef
Z3divii:
```

1.-

2.-

call __Z3divii

```
Z3divii:
LFB0:
.cfi_startproc
pushl %ebp
.cfi_def_cfa_offset 8
.cfi offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
movl 8(%ebp), %eax
cltd
idivl 12(%ebp)
popl %ebp
.cfi_restore 5
.cfi def cfa 4, 4
ret
.cfi_endproc
```

1.7 De las preguntas anteriores, se ha generado código por cada función, ambas dividen entre 4, pero difieren un poco en su implementación. Investigue a qué se debe dicha diferencia y comente cuáles podrían ser las consecuencias.

Primero existe el hecho de que además de dividir entre 4, puede dividir entre cualquier otro número, esto causa que reserve una variable extra, lo que causa una pérdida de memoria para cosas pequeñas, por otro lado, la función div4 divide únicamente entre 4, en ensamblador esta orden la toma simplemente como desplazamiento. Pero volviendo al otro caso este desplazamiento puede ser cualquier numero por lo que antes reserva una variable para esto y luego recién realiza el desplazamiento