1 a) Relocatable addresses in

main.o : .text -

- d: e8 00 00 00 00
- 12: 8b 15 00 00 00 00
- 18: 8b 05 00 00 00 00
- 20: 48 8d 3d 00 00 00 00
- 2c: e8 00 00 00 00

  .data -
- 0: 72 00

swap.o:

- 8: 48 8d 05 00 00 00 00
- f: 48 89 05 00 00 00 00
- 16: 48 8b 05 00 00 00 00
- 1f: 89 05 00 00 00 00
- 25: 48 8b 15 00 00 00 00
- 2c: 48 8b 05 00 00 00 00
- 37: 48 8b 05 00 00 00 00
- 3e: 8b 15 00 00 00 00

Other.o: .text

- 1d: c7 05 00 00 00 00 03
- 27: c7 05 00 00 00 00 04

  .data -
- 4020: 10 40 00

The locations change to:

main.o : .text-

- d: e8 00 00 00 00 -> 1156: e8 26 00 00 00
- 12: 8b 15 00 00 00 00 -> 115b: 8b 15 b3 2e 00 00
- 18: 8b 05 00 00 00 00 -> 1161: 8b 05 a9 2e 00 00
- 20: 48 8d 3d 00 00 00 00 -> 1169: 48 8d 3d 94 0e 00 00
- 2c: e8 00 00 00 00 -> 1175: e8 d6 fe ff ff

  .data -
- 0: 72 00 -> 4010: 72 00

swap.o:

- 8: 48 8d 05 00 00 00 00 -> 1189: 48 8d 05 84 2e 00 00
- f: 48 89 05 00 00 00 00 -> 1190: 48 89 05 99 2e 00 00
- 16: 48 8b 05 00 00 00 00 -> 1197: 48 8b 05 82 2e 00 00
- 1f: 89 05 00 00 00 00 -> 11a0: 89 05 72 2e 00 00
- 25: 48 8b 15 00 00 00 00 -> 11a6: 48 8b 15 83 2e 00 00
- 2c: 48 8b 05 00 00 00 00 -> 11ad: 48 8b 05 6c 2e 00 00
- 37: 48 8b 05 00 00 00 00 -> 11b8: 48 8b 05 71 2e 00 00
- 3e: 8b 15 00 00 00 00 -> 11bf: 8b 15 53 2e 00 00

Other.o: .text-

- 1d: c7 05 00 00 00 00 03 -> c7 05 1f 2e 00 00 03
- 27: c7 05 00 00 00 00 04 -> c7 05 19 2e 00 00 04

  .data -
- 4020: 10 40 00

2)

```
2
a)
 {
   0x0000000000001129 <+0>:    endbr64
   0x000000000000112d <+4>:    push   %rbp
   0x000000000000112e <+5>:    mov          %rsp,%rbp

3          int a=1,b=1;
   0x0000000000001131 <+8>:    movl   $0x1,-0x8(%rbp)
   0x0000000000001138 <+15>:    movl   $0x1,-0x4(%rbp)

4          while(a<=10)
   0x000000000000113f <+22>:    jmp          0x114f <main+38>

5          {
6          b=b*a;
   0x0000000000001141 <+24>:    mov          -0x4(%rbp),%eax
   0x0000000000001144 <+27>:    imul   -0x8(%rbp),%eax
   0x0000000000001148 <+31>:    mov          %eax,-0x4(%rbp)

7          a++;
   0x000000000000114b <+34>:    addl   $0x1,-0x8(%rbp)
--Type <RET> for more, q to quit, c to continue without paging--

4          while(a<=10)
   0x000000000000114f <+38>:    cmpl   $0xa,-0x8(%rbp)
   0x0000000000001153 <+42>:    jle0x1141 <main+24>

8          }
9          return b;
   0x0000000000001155 <+44>:    mov          -0x4(%rbp),%eax

10    }
   0x0000000000001158 <+47>:    pop          %rbp
   0x0000000000001159 <+48>:    retq
b) Local variables and their relative addresses are :
a - c7 45 f8 01 00 00 00
b - c7 45 fc 01 00 00 00

3)
a)
7    {
   0x0000000000001149 <+0>:    endbr64
   0x000000000000114d <+4>:    push   %rbp
```

```
   0x000000000000114e <+5>:    mov          %rsp,%rbp
   0x0000000000001151 <+8>:    sub          $0x20,%rsp
   0x0000000000001155 <+12>:    mov         %fs:0x28,%rax
   0x000000000000115e <+21>:    mov         %rax,-0x8(%rbp)
   0x0000000000001162 <+25>:    xor         %eax,%eax

8          struct data rec1;
9          rec1.sum=0;
   0x0000000000001164 <+27>:    movl   $0x0,-0x20(%rbp)

10         rec1.b[0]=2;
   0x000000000000116b <+34>:    movl   $0x2,-0x1c(%rbp)

11         rec1.sum=rec1.sum+rec1.b[0];
   0x0000000000001172 <+41>:    mov         -0x20(%rbp),%edx
   0x0000000000001175 <+44>:    mov         -0x1c(%rbp),%eax
   0x0000000000001178 <+47>:    add         %edx,%eax
--Type <RET> for more, q to quit, c to continue without paging--
   0x000000000000117a <+49>:    mov         %eax,-0x20(%rbp)

12         return rec1.sum;
   0x000000000000117d <+52>:    mov         -0x20(%rbp),%eax

13     }
   0x0000000000001180 <+55>:    mov         -0x8(%rbp),%rcx
   0x0000000000001184 <+59>:    xor         %fs:0x28,%rcx
   0x000000000000118d <+68>:    je          0x1194 <main+75>
   0x000000000000118f <+70>:    callq  0x1050 <__stack_chk_fail@plt>
   0x0000000000001194 <+75>:    leaveq
   0x0000000000001195 <+76>:    retq
```

b) Local variables and their relative addresses are :
 rec1.sum - c7 45 e0 00 00 00 00
 rec1.b[0] - c7 45 e4 02 00 00 00
4>
 a)
The use of the symbol main in module a1 will resolve to the declaration of main in module a1. The use of the symbol main in module a2 will resolve to the declaration of main in module a1. This is because declaration of main in a1 is a strong symbol and in a2 it is weak since it is an uninitialized global variable.

b) Linking will cause an error in part b since in module b1 main is declared as a function which is a strong symbol and in module b2 it is an initialized global variable which is also a strong symbol. This will lead to conflict and an error condition.

c) The use of the symbol main in module c1 will resolve to the declaration of main in module c1
The use of the symbol main in module c2 will resolve to the declaration of main in module c2
This is because in c1 main is a strong symbol. However on linking with c2 no error occurs since declaration of main in c2 is hidden during linking since it is a static variable.

5)
x ->

6)
a) Output of the program :
5 13
13 5
This happens because when the object file of module 2 is generated, printf statement simply prints the addresses where rec.x and rec.y are present. Since x is declared after y, the address of x is 4 bytes after y. When linking occurs, this address is overwritten by the rec struct of module 1 with {13, 5}. Hence the address y contains 13 and address x contains 5.
b) Locations that need relocation:

- d: e8 00 00 00 00
- 12: 8b 15 00 00 00 00
- 18: 8b 05 00 00 00 00
- 20: 48 8d 3d 00 00 00 00
- 2c: e8 00 00 00 00

c) The respective relocations are:

- d: e8 00 00 00 00 -> 1156: e8 26 00 00 00
- 12: 8b 15 00 00 00 00 -> 115b: 8b 15 b3 2e 00 00
- 18: 8b 05 00 00 00 00 -> 1161: 8b 05 a9 2e 00 00
- 20: 48 8d 3d 00 00 00 00 -> 1169: 48 8d 3d 94 0e 00 00
- 2c: e8 00 00 00 00 -> 1175: e8 d6 fe ff ff

7)
a)
The following error shows on compiling module 2:

```
mod2.c:5:9: error: initializer element is not constant
    5 | int z = a[3];
      |         ^
```

This is because in the 2nd module the array a has been declared as a global, its initial size should be a constant since the compiler needs to be able to allocate size for it in the executable.
b, c) The relocatable locations and their final addresses are:
main:

- d: f2 0f 10 05 00 00 00 -> 1156: f2 0f 10 05 b2 0e 00
- 15: f2 0f 11 05 00 00 00 -> 115e: f2 0f 11 05 d2 2e 00
- 1d: 48 8b 1d 00 00 00 00 -> 1166: 48 8b 1d cb 2e 00 00

- 29: e8 00 00 00 00 -> 1172: e8 2d 00 00 00
- 4a: e8 00 00 00 00 -> 1193: e8 b8 fe ff ff

fn :

- 8: 8b 05 00 00 00 00 -> 11ac: 8b 05 8a 2e 00 00