1.[10] Описати коришћене структуре података асемблера и навести у псеудокоду делове алгоритма двопролазног асемблера за обраду директиве:
а) .globl, б) .data, в) .equ
Напомене: решавати посебно сваку директиву и раздвојити први и други пролаз. Не описивати било какве структуре нити делове алгоритма које нису непосредно везани за посматрану директиву.

1st pass (.global)

```
line;
while ((line = readLine(inputFile))
    {label, dir, args} = parseLine(line)
    :
    switch dir:
        :
        case .global:
            if symTab.contains(args[0])
                symTab[args[0]].vis = glob;
            else:
                symTab.insert(newSymbol(args[0],
                    UNDEF, UNDEF, glob)
```

секција — број — глобалност

2nd pass
— nothing —

1 st pass (.data)

```
sections = {}
currentSec = null
locationCounter = 0

while (line = getLine(inputFile))
    {label, dir, args} := parseLine(line)
    switch dir:
        :
        case .data:
            sections.append(currentSection)
            currentSection = new Section(".data")
            locationCounter = 0;
        :
end pass.

sections = {}
currentSec = null
locationCounter = 0
sectionContent = ... (void*)
while (line = getLine(inputFile))
    {label, dir, args} := parseLine(line)
    switch dir:
        :
        case .data:
            currentSection.setContent(sectionContent)
            currentSection = sections.get(".data")
            locationCounter = 0;
```

1 st pass (.eqn)

```
line;
while ((line = readLine(inputFile))
    {label, dir, args} = parseLine(line)
    switch dir:
        :

        case .eqn:
            name = args[0]
            expr = args[1]
            if symTab.contains(name):
                error("already defined")
            else:
                sym = new Symbol(name, )
                callasOf(expr), valueOf(expr), local()
                symTab.insert(sym);
        :

2 nd pass
— nothing —
```

1.[10] а) Навести један пример доделе и употребе релокативне вредности симболу у програму на симболичком машинском језику x86 и један пример доделе и употребе апсолутне вредности симболу у истом програму. Поред одговарајућих инструкција/директива исписати коментар "додела" или "употреба".

б) Објаснити на претходном примеру како асемблер доставља пуниоцу информације о релокативним вредностима у програму, ако се користи ELF формат објектног програма.

ц) Објаснити на који начин пунилац обрађује релокативне вредности у програму према подацима из ELF објектног програма.

```
.section .text
foo :
bar :
```

```
.equ A, foo + 5        ← додела рел. вредности
.equ B, bar - foo      ← додела апс. вредности
```

```
.section .data
.long A                ← употреба рел. вредн.
.long B.               ← употреба апс. вредн.
```

Кроз секцију .symtab чији је садржај низ структура који описују симболе

```
typedef struct {
    uint32_t      st_name;
    unsigned char st_info;
    unsigned char st_other;
    uint16_t      st_shndx;
    Elf64_Addr    st_value;
    uint64_t      st_size;
} Elf64_Sym;
```