a.

<идентификатор> ::= ("\_" | <буква > ){ <буква> | <цифра> | "\_" }

<буква> ::= "A" | "B" | ... | "Z" | "a" | "b" | ... | "z"

<цифра> ::= "0" | "1" | ... | "9"

G=(T,N,P,S)

T={("1","2",…"9", "A", "B",…"Z","a","b",…,"z","\_"} (терминальные символы)

N={I,L,N,F,X} (нетерминальные символы)

S — начальный символ

Правила P:

1. S 🡪 F|FX
2. F 🡪 L|"\_"
3. X 🡪 LX|NX|"\_"X| ε
4. L 🡪 "A" | "B" | ... | "Z" | "a" | "b" | ... | "z"
5. N 🡪 "0" | "1" | ... | "9"

А) Приведите пример вывода правильной цепочки языка.

\_Num\_1

Б) Постройте дерево вывода для этой цепочки (левосторонний и правосторонний вывод)

1 S → FX → \_X → \_XL → \_XLL → \_XLLL → \_XLLL\_ → \_XLLL\_N → \_NLLL\_N → \_NuLL\_N → \_NumL\_N → \_Num\_N → \_Num\_1

2 S → FX → FX → FXN → FX1 → FX\_1 → FXL\_1 → FXm\_1 → FXLm\_1 → FXum\_1 → FNum\_1.

Каждое правило вывода в этой грамматике имеет вид, который соответствует **контекстно-свободной грамматике** (тип 2), где нетерминал стоит слева от последовательности, справа - последовательность терминалов и нетерминалов без контекстных ограничений.

