Variant 1

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, B, D, A\}$, $V_T=\{a,b,c,d\}$,
$P=\{$ 1. $S \rightarrow d\,B$ 2. $B \rightarrow D$ 3. $B \rightarrow D\,c\,B$
4. $D \rightarrow b\,A$ 5. $A \rightarrow a$ 6. $A \rightarrow a\,A$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***dbacbaa*** string
Bonus point: provide as an additional output the derivation tree.

Variant 2

2. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, B, D, A\}$, $V_T=\{a,b,c,d\}$,
$P=\{$ 1. $S \rightarrow d\,A$ 2. $A \rightarrow D$ 3. $A \rightarrow D\,c\,A$
4. $D \rightarrow b\,B$ 5. $B \rightarrow a$ 6. $B \rightarrow a\,B$ $\}$,
implement the LL(1) parsing and analyze the ***dbaacbaaa*** string
Bonus point: provide as an additional output the derivation tree.

Variant 3

For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d\}$,
$P=\{$ 1. $S \rightarrow A$ 2. $A \rightarrow c\,B$ 3. $B \rightarrow C\,d$
4. $C \rightarrow D$ 5. $C \rightarrow C\,b\,D$ 6. $D \rightarrow a$ 7. $D \rightarrow a\,c\,C\,d$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***cabacadd*** string
Bonus point: provide as an additional output the derivation tree.

Variant 4

2. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,e\}$,
$P=\{$ 1. $S \rightarrow C$ 2. $C \rightarrow B$ 3. $C \rightarrow B\,e\,C$
4. $B \rightarrow a\,b\,D$ 5. $D \rightarrow A\,d$ 6. $A \rightarrow c$ 7. $A \rightarrow A\,c$ $\}$,
implement the LL(1) parsing and analyze the ***abcdeabcccd*** string
Bonus point: provide as an additional output the derivation tree.

Variant 5

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,e\}$,
$P=\{$ 1. $S \rightarrow A\,e$ 2. $A \rightarrow b\,a\,B$ 3. $B \rightarrow C\,d$
4. $C \rightarrow D$ 5. $C \rightarrow C\,b\,D$ 6. $D \rightarrow c$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***bacbcbcde***
Bonus point: provide as an additional output the derivation tree.

Variant 6

2. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d\}$,
$P=\{$ 1. $S \rightarrow d\,B$ 2. $B \rightarrow C$ 3. $B \rightarrow C\,c\,B$
4. $C \rightarrow b\,A$ 5. $A \rightarrow a$ 6. $A \rightarrow a\,A$ $\}$,
build the LL(1) parse table and analyze the ***dbacbaaa*** string
Bonus point: provide as an additional output the derivation tree.

Variant 7

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C\}$, $V_T=\{a,b,c\}$,
$P=\{$ 1. $S \rightarrow aA$ 2. $A \rightarrow C$ 3. $A \rightarrow C\,b\,A$
4. $C \rightarrow d\,B$ 5. $B \rightarrow b$ 6. $B \rightarrow a\,B$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***adbbdb*** string
Bonus point: provide as an additional output the derivation tree.

Variant 8

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,e\}$,
$P=\{$ 1. $S \rightarrow L\,d\,X$ 2. $X \rightarrow D$ 3. $L \rightarrow c\,a$
4. $L \rightarrow aL$ 5. $D \rightarrow b$ 6. $D \rightarrow D\,e\,b$ $\}$,
implement the LL(1) parsing and analyze the ***aaaacadeebbb*** string
Bonus point: provide as an additional output the derivation tree.

Variant 9

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,e\}$,
$P=\{$ 1. $S \to C$ 2. $C \to B\ c\ A$ 3. $A \to b$
4. $A \to d\ D$ 5. $D \to B\ e$ 6. $B \to a$ 7. $B \to B\ b\ a$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***abacdae*** string
Bonus point: provide as an additional output the derivation tree.

Variant 10

2. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,e\}$,
$P=\{$ 1. $S \to A$ 2. $A \to C$ 3. $A \to A\ c\ C$
4. $C \to a$ 5. $C \to b$ 6. $C \to d\ D$ 7. $D \to A\ e$ $\}$,
implement the LL(1) parsing and analyze the ***dacbcbeca*** string
Bonus point: provide as an additional output the derivation tree.

Variant 11

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,e\}$,
$P=\{$ 1. $S \to B$ 2. $B \to C$ 3. $B \to C\ c\ B$
4. $C \to a\ d\ D$ 5. $D \to A\ e$ 6. $A \to b$ 7. $A \to A\ b$, 8. $B \to d$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***adabcd*** string
Bonus point: provide as an additional output the derivation tree.

Variant 12

2. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,e,f,g\}$,
$P=\{$ 1. $S \to C\ f$ 2. $C \to a\ b\ B$ 3. $B \to D\ c$
4. $D \to A$ 5. $D \to g\ A$ 6. $A \to d$ 7. $A \to e$ $\}$,
implement the LL(1) parsing and analyze the ***abgdcf*** string
Bonus point: provide as an additional output the derivation tree.

Variant 13

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C\}$, $V_T=\{a,b,c,d,e, f\}$,
$P=\{$ 1. $S \to B\ c$ 2. $S \to B\ c\ d\ C$ 3. $C \to A\ e$
4. $A \to f$ 5. $A \to A\ b\ f$ 6. $B \to a$ 7. $B \to B\ b\ a$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***abacdfbfe*** string
Bonus point: provide as an additional output the derivation tree.

Variant 14

2. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,e,f\}$,
$P=\{$ 1. $S \to A\ a$ 2. $S \to A\ a\ b\ B$ 3. $B \to C\ c$
4. $C \to f$ 5. $C \to C\ d\ f$ 6. $A \to e$ 7. $A \to A\ d\ e$ $\}$,
implement the LL(1) parsing and analyze the ***ededeabfdfc*** string
Bonus point: provide as an additional output the derivation tree.

Variant 15

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,e,f,g,e\}$,
$P=\{$ 1. $S \to A\ g$ 2. $A \to a\ b\ c\ B$ 3. $B \to C\ d$
4. $C \to e$ 5. $C \to C\ f\ D$ 6. $D \to e$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***abcefedg*** string
Bonus point: provide as an additional output the derivation tree.

Variant 16

2. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, B, D, A\}$, $V_T=\{a,b,c,d\}$,
$P=\{$ 1. $S \to d\ A$ 2. $A \to B$ 3. $A \to B\ c\ A$
4. $B \to b\ D$ 5. $D \to a$ 6. $D \to a\ D$ $\}$,
implement the LL(1) parsing and analyze the ***dbaacbaaa*** string

Bonus point: provide as an additional output the derivation tree.

Variant 17

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, B, D, A\}$, $V_T=\{a,b,c,d\}$,
$P=\{$ 1. $S \to d\,B$ 2. $B \to a$ 3. $B \to a\ A$
4. $A \to D$ 5. $A \to D\,c\,A$ 6. $D \to b\,B$ 7. $A \to c$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***dabacba*** string
Bonus point: provide as an additional output the derivation tree.

Variant 18

2. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,e\}$,
$P=\{$ 1. $S \to A$ 2. $A \to B$ 3. $A \to B\,e\,A$
4. $B \to a\,b\,D$ 5. $D \to C\,d$ 6. $C \to c$ 7. $C \to C\,c$ $\}$,
implement the LL(1) parsing and analyze the ***abcdeabcccd*** string
Bonus point: provide as an additional output the derivation tree.

Variant 19

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,e,f,g,e\}$,
$P=\{$ 1. $S \to A\,g$ 2. $A \to A\,b\,D$ 3. $A \to C$
4. $C \to e$ 5. $C \to C\,f\,D$ 6. $D \to e$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***efefebeg*** string
Bonus point: provide as an additional output the derivation tree.

Variant 20

2. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, D\}$, $V_T=\{a,b,c,d\}$,
$P=\{$ 1. $S \to A\,a$ 2. $S \to A\,a\,d\,B$ 3. $B \to D\,a\,f$
4. $D \to c$ 5. $D \to D\,e\,c$ 6. $A \to b$ 7. $A \to A\,e\,b$ $\}$,
implement the LL(1) parsing and analyze the ***bebadcecaf*** string
Bonus point: provide as an additional output the derivation tree.

Variant 21

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, F, L, E\}$, $V_T=\{a,b,c,d,e\}$,
$P=\{$ 1. $S \to L\,d\,F$ 2. $F \to E$ 3. $L \to c\,a$
4. $L \to L\,a$ 5. $E \to b$ 6. $E \to E\,e\,b$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***caadbeb*** string
Bonus point: provide as an additional output the derivation tree.

Variant 22

2. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C, D\}$, $V_T=\{a,b,c,d,f,g\}$,
$P=\{$ 1. $S \to B\,f\,D$ 2. $B \to B\,c$ 3. $B \to D$
4. $D \to A\,g$ 5. $D \to A$ 6. $A \to d$ 7. $A \to c$ $\}$,
implement the LL(1) parsing and analyze the ***dgcfdg*** string
Bonus point: provide as an additional output the derivation tree.

Variant 23

1. For the given grammar $G=(V_N, V_T, P, S,)$, $V_N=\{S, A, B, C\}$, $V_T=\{a,b,c,d,e, f\}$,
$P=\{$ 1. $S \to B\,c$ 2. $S \to A\ c\,d\,C$ 3. $C \to C\,e$
4. $C \to f$ 5. $A \to B\,b\,f$ 6. $B \to a$ 7. $B \to B\,b\,a$ $\}$,
implement algorithm of simple precedence parsing and analyze the ***ababfcdf*** string
Bonus point: provide as an additional output the derivation tree.

Variant 24

2. For the given grammar G=(V_N, V_T, P, S,), V_N ={S, A, B, C, D}, V_T ={a,b,c,d,e,f},
P={ 1. $S \rightarrow D\ a\ b\ B$ 2. $D \rightarrow e$ 3. $D \rightarrow D\ d\ f$
4. $B \rightarrow A\ c\ D$ 5. $A \rightarrow a$ 6. $A \rightarrow A\ d\ e$ },
implement the LL(1) parsing and analyze the ***edfabadece*** string
Bonus point: provide as an additional output the derivation tree.

Variant 25

1. For the given grammar G=(V_N, V_T, P, S,), V_N ={S, A, B, C, D}, V_T ={a,b,c,d,e,f,g,e},
P={ 1. $S \rightarrow A\ g\ C$ 2. $A \rightarrow a\ B$ 3. $B \rightarrow C$,
4. $D \rightarrow D\ a$ 5. $D \rightarrow e$ 6. $C \rightarrow C\ f\ D$, 7. $C \rightarrow d$ },
implement algorithm of simple precedence parsing and analyze the ***adfegdfe*** string
Bonus point: provide as an additional output the derivation tree.

Variant 26

2. For the given grammar G=(V_N, V_T, P, S,), V_N ={S, F, L, E}, V_T ={a,b,c,d,e},
P={ 1. $S \rightarrow E\ d\ F$ 2. $E \rightarrow Fe\ b$ 3. $F \rightarrow F\ a\ L$
4. $F \rightarrow b$ 5. $L \rightarrow ca$ 6. $L \rightarrow L\ a$ },
implement the LL(1) parsing and analyze the ***bacaebdbaca*** string
Bonus point: provide as an additional output the derivation tree.

Variant 27

1. For the given grammar G=(V_N, V_T, P, S,), V_N ={S, F, L, E}, V_T ={a,b,c,d,e},
P={ 1. $S \rightarrow E\ d\ F$ 2. $E \rightarrow e\ bD$ 3. $F \rightarrow F\ a\ L$
4. $D \rightarrow FbE$ 5. $L \rightarrow aL$ 6. $L \rightarrow a$ },
implement algorithm of simple precedence parsing and analyze the ***bacaebdbaca*** string
Bonus point: provide as an additional output the derivation tree.