1.1

We label each rule from top to bottom 1 - 6.

We see that rule 3:

$$B \to B\ ;\ +\ V \mid +\ V$$

Is left-recursive as the B in the production can keep expanding infinitely without consuming anything. We remove the left recursion with the following modification:

$$B\ \to +\ V\ ;\ B \mid\ +\ V$$

This gives rise to a new problem, the common prefix $+$. When we consume a $+$ we have no way of knowing which rule to follow next (i.e. if we should add a $V\ ;\ B$ or only a V). To fix this, we perform left factoring by factoring the parts after $+\ V$ out of B:

$$\begin{aligned} B &\to\ +\ V\ I \\ I &\to\ ;\ B \mid \epsilon \end{aligned}$$

No other rule has left recursion, and consequently the grammar is now LL(1) parsable.The full grammar with numbered productions becomes:

1. $S \to i\ C\ B\ E\ F\ n$
2. $C \to c$
3. $B \to +\ V\ \ I$
4. $I \to ;\ B$
5. $I \to \epsilon$
6. $V \to x$
7. $V \to y$
8. $E \to e\ B$
9. $E \to \epsilon$
10. $F \to f$
11. $F \to \epsilon$

1.2

Non-terminal	FIRST	FOLLOW	NULLABLE
S	i_1	$\$$	NO
C	c_2	$+$	NO
B	$+_3$	$e,\ f,\ n$	NO
I	$;_4\ \epsilon_5$	$e,\ f,\ n$	YES
V	$x_6\ y_7$	$;\ ,\ e,\ f,\ n$	NO
E	$e_8\ \epsilon_9$	$f,\ n$	YES
F	$f_{10}\ \epsilon_{11}$	n	YES

1.3

LL(1) table:

	i	c	$+$	$;$	x	y	e	f	n
S	1								
C		2							
B			3						
I				4			5	5	5
V					6	7			
E							8	9	9
F								10	11

No cell has more than one number (rule), so there are no conflicts → the grammar can be parsed with an LL(1) algorithm. Moreover, the grammar is unambiguous.