Name : Vishal Reddy Mandadi Roll Number:2019101119

Note : Here in this pdf I assigned each line of code with a number and thereby used them while referring to corresponding line of code as code-line <line_number>

Step by step explanation to the following code is as follows :

1. int x = 2019101119 % 100 ;
2. int a = -1 * (x) ;
3. unsigned int b = (unsigned int) a;
4. unsigned int c = UINT_MAX – x ;
5. int d = (int) c ;
6. int p = 65490 + x ;
7. short int e = (short int) p ;
8. unsigned short f = (unsigned short) a;
9. printf (“ %d %u %u %d %hu %hu \n ” , a , b , c , d , e , f) ;

OUTPUT = -19 4294967277 4294967276 -20 -27 65517

a) In the code-line (1) I had replaced <your_rollnumber> with 2019101119,. Then x=2019101119%100 =19 (since 2019101119%100 = (2019101100+19)%100=19%100=19)
   - ⇨ x=19
   - ⇨ signed two’s complement representation of 19 is = 00000000000000000000000000010011.
b) In code-line (2) ‘a’ becomes equal to -19 (trivial), signed two’s complement representation of (-19) is = 11111111111111111111111111101101.
c) In code-line (3) ‘a’ is being type casted from int to unsigned int. So here a’s value won’t be altered but it is converted into unsigned form before being stored into ‘b’.(Note that the binary value is still 11111111111111111111111111101101 but the way it will be interpreted will be different based on signed or unsigned type definition)
d) So now b =11111111111111111111111111101101 in binary which when converted into decimal format (noting that binary representation is now unsigned) will be equal to 4294967277, => b=4294967277

e) (In code-line (4))UINT_MAX returns max value that can be stored by an unsigned int data type. It's definition is included in <limits.h> header file. UINT_MAX=4294967295. So UINT_MAX -x = 4294967295-19=4294967276, therefor c=4294967276

f) Now d(code-line (5)) takes the same binary words of c but interprets it as signed two's complement form(due to type casting without affecting c).
This implies binary quad word of
d =11111111 11111111 11111111 11101100
but when converted into decimal , d= -20 .

g) Now in code-line (6) `p= 65490 + x = 65490 + 19=65509`
Since int max value is greater than 65509 hence p = 65509(no overflow condition)

h) Signed two's complement form of
p's value=00000000 00000000 11111111 11100101, now coverting p's value into short datatype leads to truncation of first 16 bits(counting from MSB) => value of e = 11111111 11100101 in binary form which when interpreted as short int e=-27 (because 2's complement of 11111111 11100101 is 11011 whose value is 27 and MSB is 1 here so value becomes (-27))

f) We know that value of 'a' is 65509 , and max value that short can store is 32767 and unsigned short int can store 65535. Since 65509 < 65535 , no overflow occurs and hence value of f remains = 65509.

i) Finally printf command has various attributes like %u, %d ,.. where each of them suggests the interpretation that compiler must follow before generating the respective output