1.Integer Representation

The output of given code snippet:

-4 4294967292 4294967291 -5 -42 65532

Code Snippet:

```
1       int x = your_rollnumber %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 %hi %hu\n” , a , b , c , d , e , f ) ;
```

In 1st line, int x = 2019101104 % 100
so, x = 4

In 2nd line, int a = -4
so, a = -4

In 3rd line, we are typecasting int ‘a’ to unsigned int. In 2’s complement representation
-4 is 1111 1111 1111 1111 1111 1111 1111 1100 (32 bits)
So the range is 0 to (2^32)-1 i.e., 0 to 4294967295.
From this if we remove the last two rightmost places in binary number which is zero
It becomes 4294967295 - 3 = 4294967292
So, b = 4294967292

In 4th line, UINT_MAX is the maximum value that can be stored in integer data type.
UINT_MAX = (2^32) – 1 = 4294967295
# UINT_MAX can be found in <limits.h>
UINT_MAX – x = 4294967295 – 4 = 4294967291
So, c = 4294967291

In 5th line, we are typecasting unsigned int ‘c’ to int. In binary representation,
4294967291 is 1111 1111 1111 1111 1111 1111 1111 1011
So, If we convert to int, it is in Two’s complement representation, and all the significant bits are ‘1’ which implies it is a negative number, So to get the magnitude we are inverting all the bits and adding ‘1’ , it becomes
0000 0000 0000 0000 0000 0000 0000 0101 that is 5
So, d = -5

In 6th line, int p = 65490 + x = 65490 + 4 = 65494
So, p = 65494

In 7th line, we are typecasting int ‘p’ to short int. In binary representation,
65494 is 1111 1111 1101 0110
It is exactly 2 bytes which is size of short int. When we are converting int to short int , we have ‘1’ in left most significant bit which indicates a negative number

We get, e = -(2^16) + (2^15) +(2^14) + ... + (2^6) + (2^4) + (2^2) + (2^1) = -42

So e = -42

In 8th line, we are typecasting int ‘a’ to insigned short. In Two’s complement

- 4 = 1111 1111 1111 1100 (as it is now unsigned short ie. 2 bytes)

we get, f = (2^16) – 1 – 1 – 2 = 65535 – 3 = 65532

So, f = 65532