

# 2s Compliment Calculator

## Example:

```
C:\WINDOWS\py.exe  X  +  v
Enter decimal value: -127
Enter Total Bit Length (including sign bit): 8

-----
1. Convert to Binary
If negative decimal, change to positive decimal into 7-bit binary first
Decimal:          -127
+ve Decimal:      127
Binary:           01111111
removesignbit:    1111111

-----
2. Inverting the bit
invert binary:     0000000

-----
3. Add 1 complement
+1 Binary:         0000001

-----
4. Apply negative sign to the MSB
2scomplement:     10000001

-----
Answer: -127 = 10000001
-----

Enter decimal value:
```

## Code:

```
def binaryToDecimal(val: str) -> int:
    return int(val, 2)

def decimalToBinary(n: int) -> str:
```

```
return bin(n).replace("0b", "")
```

```
def fillBinaryZero(binary: str, bitLenght: int) -> str:
```

```
    offset = bitLenght - len(binary)
```

```
    toReturn = binary[::-1]
```

```
    for i in range(0,offset):
```

```
        toReturn += '0'
```

```
    return toReturn[::-1]
```

```
def twosComplement(binary: str, isNegative: bool) -> str:
```

```
    # 1. Convert the positive value into 7-bit binary
```

```
    # remove MSB
```

```
    reversedStrBinary = binary[1:][::-1]
```

```
    print('removesignbit: ', reversedStrBinary[::-1])
```

```
    # Invert
```

```
    # 2. Invert the digits (12 -> 02, 02 -> 12)
```

```
    invertedStrBinary = ""
```

```
    for i in range(0,len(reversedStrBinary)):
```

```
        invertedStrBinary += '0' if reversedStrBinary[i] == '1' else '1'
```

```
    print("\n-----")
```

```
    print('2. Inverting the bit')
```

```
    print('invert binary: ', invertedStrBinary[::-1])
```

```
    # add 1s
```

```
    # 3. Add 1s complement
```

```
    if '1' in invertedStrBinary:
```

```
        invertedStrBinary = fillBinaryZero(decimalToBinary(binaryToDecimal(invertedStrBinary[::-1]) + 1),
```

```
len(reversedStrBinary))
```

```
    else:
```

```
        invertedStrBinary = invertedStrBinary[0:-1] + '1'
```

```
    print("\n-----")
```

```
    print('3. Add 1 complement')
```

```
    print('+1 Binary: ', invertedStrBinary)
```

```
    # 4. Add 1s sign bit to negative
```

```
    print("\n-----")
```

```
    print('4. Apply negative sign to the MSB')
```

```
    return '1' + str(invertedStrBinary)
```

```

def main() -> None:
    """ Description """

    while True:
        userInput = input("Enter decimal value: ")

        # Without any input, break the loop
        if len(userInput) == 0:
            print("End of program.")
            break

    try:
        while True:
            bitLenght = int(input("Enter Total Bit Lenght (including sign bit): "))
            break
    except:
        print("Invalid bit lenght")
    decimal = int(userInput)

    if bitLenght - len(userInput) < 0:
        print('Invalid Bit Lenght')
    else:
        # check if negative
        if decimal < 0:
            isNegative = True
            decimal = decimal * -1
        else:
            isNegative = False

    binary = fillBinaryZero(str(decimalToBinary(decimal)), bitLenght)
    print('\n-----')
    print('1. Convert to Binary\nIf negative decmial, change to positive decimal into 7-bit binary first')
    print(f'Decimal: {userInput:>14}\n+ve Decimal: {decimal:>10}\nBinary: {binary:>15}')
    finalBinary = twosComplement(binary,isNegative) if isNegative else binary
    print('2scomplement: ', finalBinary)
    print('\n-----')
    print(f'Answer: {userInput} = {finalBinary}')
    print('-----\n')

```

```
if __name__ == "__main__":  
    main()
```

---

Revision #2

Created 1 January 2024 06:42:52 by aki

Updated 1 January 2024 06:45:17 by aki