```
#!/usr/bin/env python3

import sys

if len(sys.argv) == 1:
        script_name = sys.argv[0]
        print("Sintax: " + script_name + " samfile genomelength")
        sys.exit()
elif len(sys.argv) == 2:
        sys.exit("Sintax: One parameter [samfile or genomelength] is missing")
elif len(sys.argv) > 2:

        #initialize genome_change variable as a list constituted by 0 with length
= genomelength
        genome_length = int(sys.argv[2])
        genome_change = [0]*genome_length

        #open sam file
        sam_file = open(sys.argv[1], 'r')

        #read each line of the file
        for line in sam_file:
                # if we want to remove whitespace characters
                line = line.rstrip()
                # if line starts with @ we have to skip this line
                if line.startswith('@'):
                        continue

                # creation of a list containing all columns of each row. Row is
splitted by tab = \t
                fields = line.split("\t")

                # flag indicates that both reads align correctly
                #if ((fields[1] & 3) == 3) and (fields[8] > 0): # => bitwise
operation

                # convertion of flag from integer to binary
                flag = bin(int(fields[1]))
                # get the last two number of flag
                interesting_flag = flag[-2:]
                starting_mate_position = int(fields[3])
                mate_length = int(fields[8])
                #if (interesting_flag == '11') and (mate_length > 0):
                if flag.endswith('11') and (mate_length > 0):
                        # increment start position by one
                        genome_change[starting_mate_position] += 1
                        # decrement end position by one
                        genome_change[starting_mate_position + mate_length] -= 1

        # print genomic profile as a wiggle file
        sam_file.close()
        print("fixedStep chrom=genome start=1 step=1 span=1")
        current_coverage = 0
        # cicle over all positions of the genome
        for position in range(genome_length):
                current_coverage += genome_change[position]
                print(current_coverage)
```