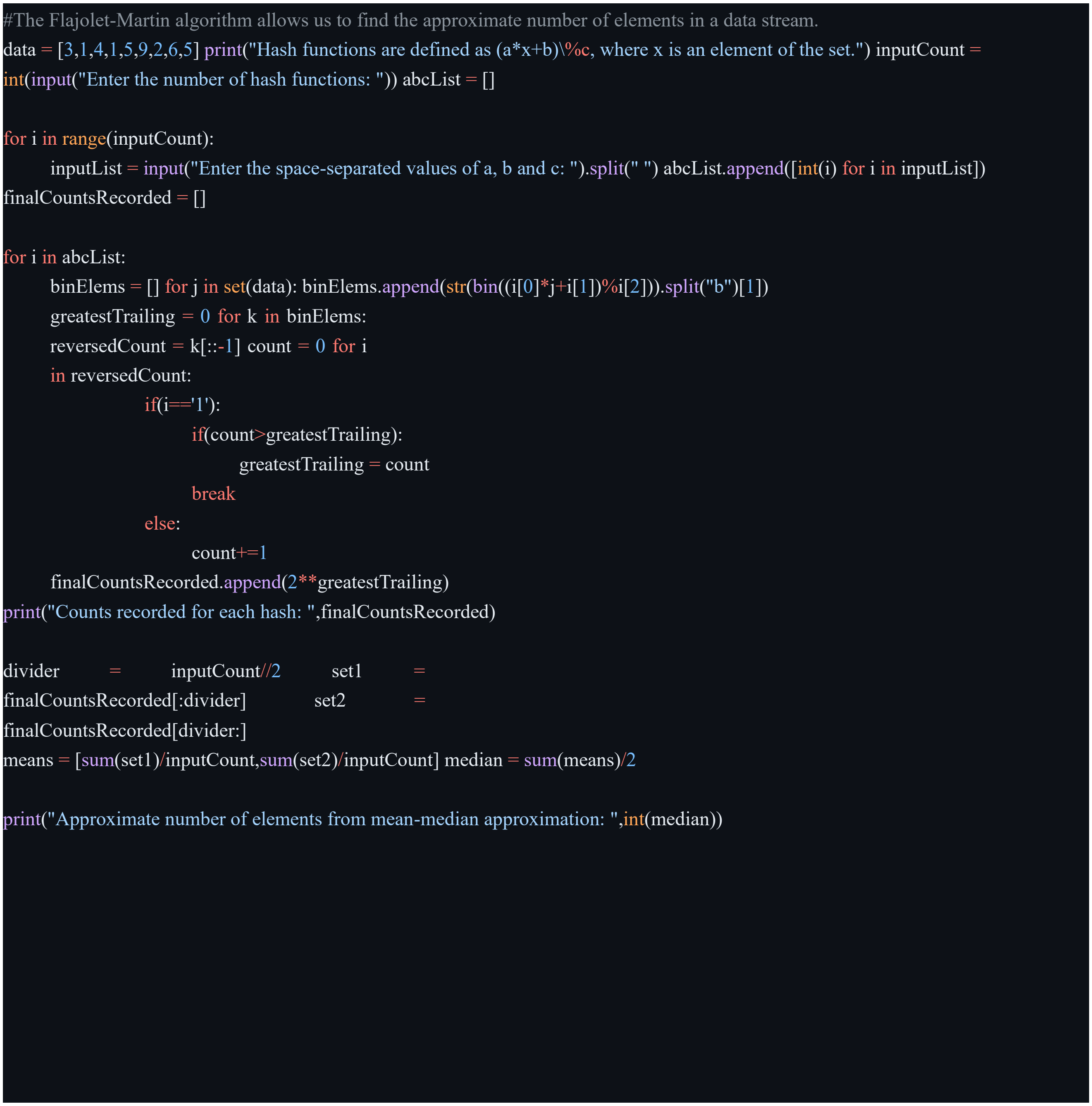
EXPERIMENT - 7

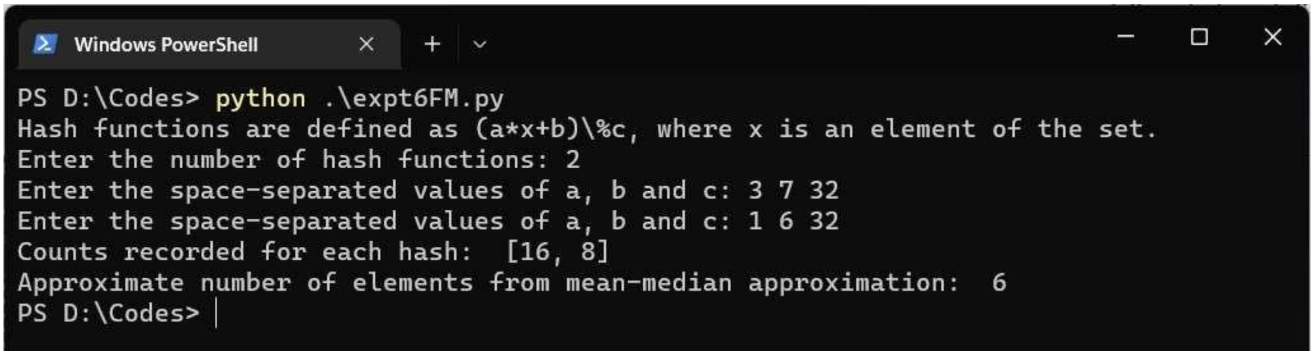
AIM:- Implement Flajolet Martin algorithm using any programming language.

CODE:-



```
#The Flajolet-Martin algorithm allows us to find the approximate number of elements in a data stream.
data = [3,1,4,1,5,9,2,6,5] print("Hash functions are defined as (a*x+b)\%c, where x is an element of the set.") inputCount =
int(input("Enter the number of hash functions: ")) abcList = []

for i in range(inputCount):
        inputList = input("Enter the space-separated values of a, b and c: ").split(" ") abcList.append([int(i) for i in inputList])
finalCountsRecorded = []

for i in abcList:
        binElems = [] for j in set(data): binElems.append(str(bin((i[0]*j+i[1])%i[2])).split("b")[1])
        greatestTrailing = 0 for k in binElems:
        reversedCount = k[::-1] count = 0 for i
        in reversedCount:
                        if(i=='1'):
                                if(count>greatestTrailing):
                                        greatestTrailing = count
                                break
                        else:
                                count+=1
        finalCountsRecorded.append(2**greatestTrailing)
print("Counts recorded for each hash: ",finalCountsRecorded)

divider        =        inputCount//2        set1        =
finalCountsRecorded[:divider]           set2           =
finalCountsRecorded[divider:]
means = [sum(set1)/inputCount,sum(set2)/inputCount] median = sum(means)/2

print("Approximate number of elements from mean-median approximation: ",int(median))
```

OUTPUT:-



```
PS D:\Codes> python .\expt6FM.py
Hash functions are defined as (a*x+b)\%c, where x is an element of the set.
Enter the number of hash functions: 2
Enter the space-separated values of a, b and c: 3 7 32
Enter the space-separated values of a, b and c: 1 6 32
Counts recorded for each hash:  [16, 8]
Approximate number of elements from mean-median approximation:  6
PS D:\Codes>
```