In [15]:
```
import numpy as np
import math
import matplotlib.pyplot as plt
import pandas as pd
from numpy import random
```

Part 1A: Multiple Choice questions

the following cells contain the probibilty to answer correctly 12 questions containing 5 choices with one correct answer each

In [3]:
```
Qnum = 12
Qselect = 5
Qcorrect = 1

print(math.comb(Qnum,Qselect))
```

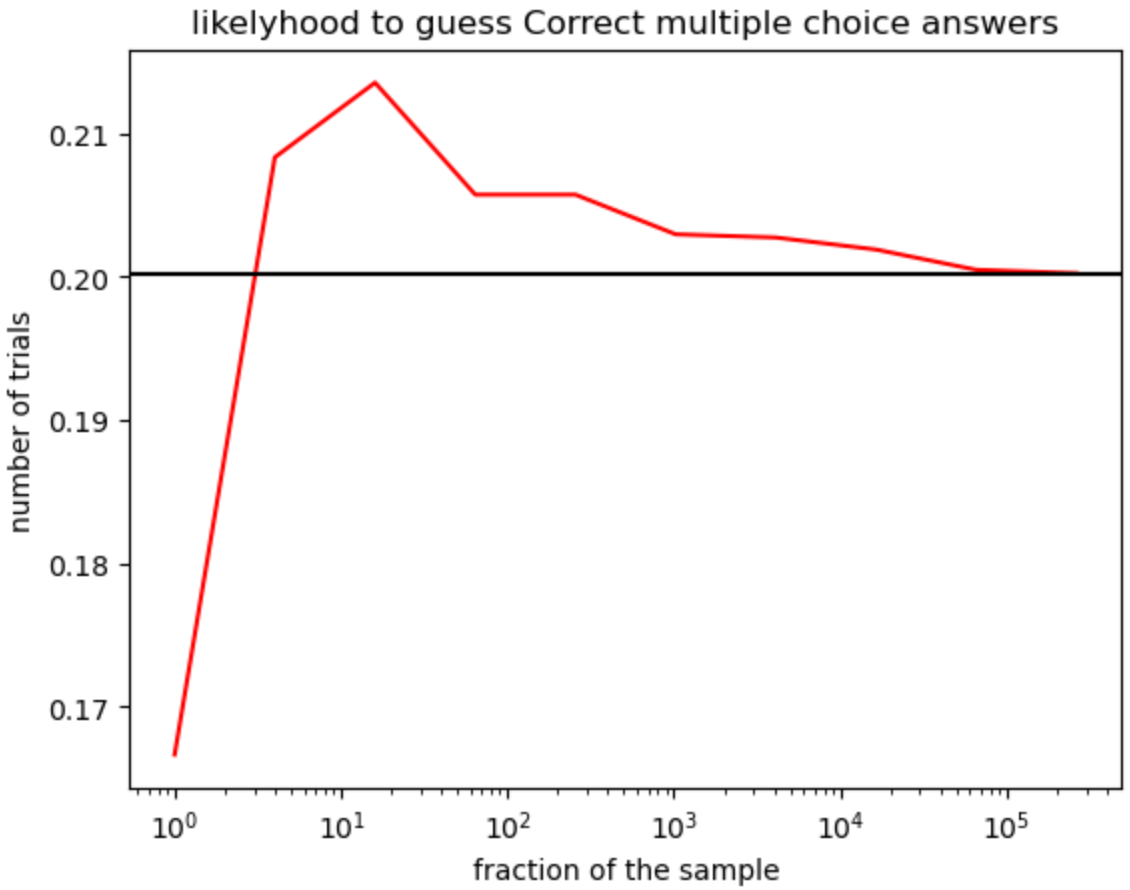
```
792
```

In [5]:
```
NumList = [pow(4,k) for k in range(10)]

samplesize = NumList[np.shape(NumList)[0]-1]
ProbList = []
CorrectAnsTrial = 0
CorrectAnsTot = 0
totalruns = samplesize*Qnum
count = 0

for i in range(samplesize+1):
  CorrectAnsTrial = 0
  for k in range(Qnum):
    ans = random.randint(Qselect)
    if ans == 0:
      CorrectAnsTrial = CorrectAnsTrial + 1
  CorrectAnsTot = CorrectAnsTot + CorrectAnsTrial

  if i == NumList[count]:
    ProbList.append(CorrectAnsTot/(NumList[count]*Qnum))
    count = count + 1

print(CorrectAnsTot/totalruns)
# print(np.shape(ProbList))
```

```
# print(ProbList)
print(NumList[np.shape(NumList)[0]-1])
```

```
0.20024522145589194
262144
```

In [6]:
```
plt.plot(NumList, ProbList,'r')
plt.axhline(y= CorrectAnsTot/totalruns, color='black', linestyle='-')
plt.ylabel('number of trials')
plt.xlabel('fraction of the sample')
plt.title('likelyhood to guess Correct multiple choice answers')
plt.xscale('log')
plt.show()
```



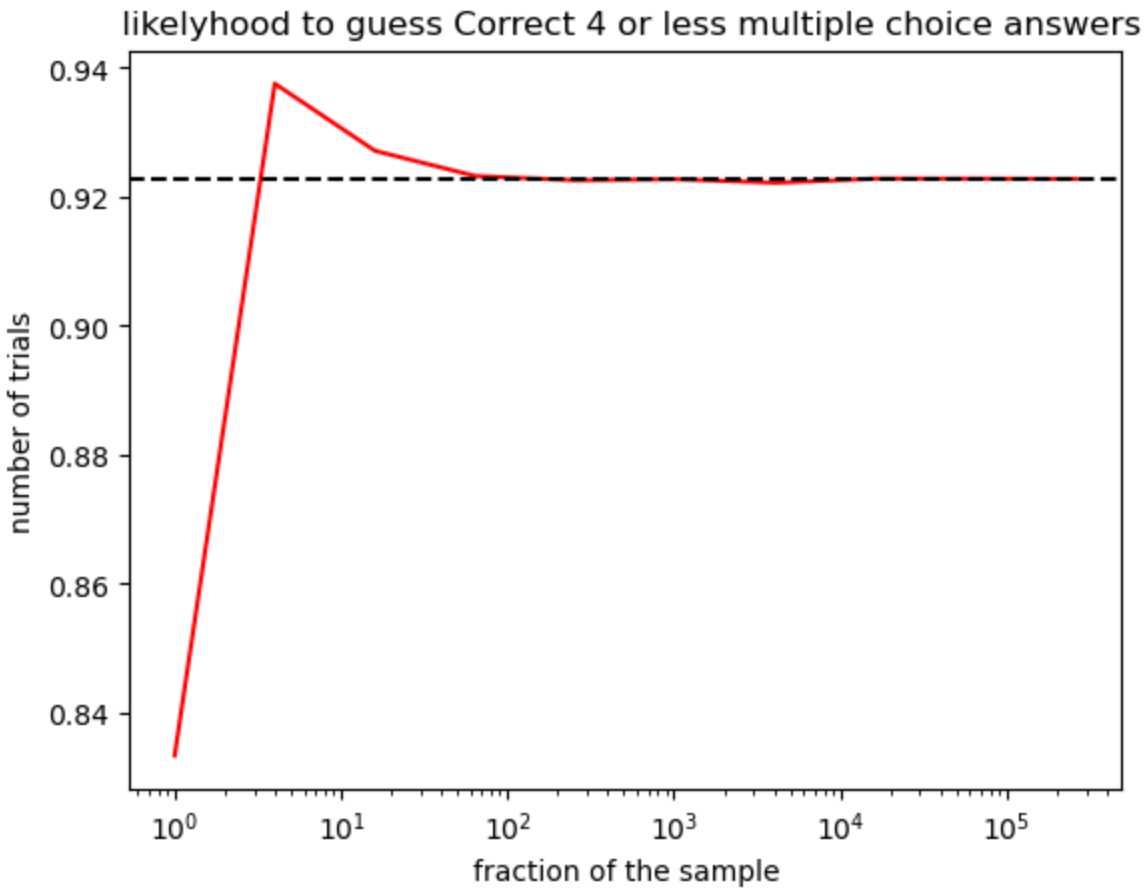
Part 1b

find the probability to have 4 or less (of 12) questions answered correctly

In [8]:
```
NumList = [pow(4,k) for k in range(10)]

samplesize = NumList[np.shape(NumList)[0]-1]
ProbList = []
CorrectAnsTrial = 0
CorrectAnsTot = 0
totalruns = samplesize*Qnum
count = 0

for i in range(samplesize+1):

  CorrectAnsTrial = 0
  for k in range(Qnum):
    ans = random.randint(Qselect)
    if ans == 0:
      CorrectAnsTrial = CorrectAnsTrial + 1

  if CorrectAnsTrial <= 4:
    CorrectAnsTot = CorrectAnsTot + 1

  if i == NumList[count]:
    ProbList.append(1 - CorrectAnsTot/(NumList[count]*Qnum))
    count = count + 1

print(1-CorrectAnsTot/totalruns)
```

```
0.9227244059244791
```

In [9]:
```
plt.plot(NumList, ProbList,'r')
plt.axhline(y= 1 -CorrectAnsTot/totalruns, color='black', linestyle='--')
plt.ylabel('number of trials')
plt.xlabel('fraction of the sample')
plt.title(' likelyhood to guess Correct 4 or less multiple choice answers')
plt.xscale('log')
plt.show()
```



Part 2

with replacment

In [11]:
```
runs = 10000

Select = 10
Population = 1000
NumSmoke = 350
PopList = [*[0 for k in range(Population-NumSmoke)],*[ 1 for k in range(NumSmoke)]]

CountSmoke = 0
CountSmokeTot = 0
```

```
SmokePlot =[]

for i in range(runs):
  CountSmoke = 0
  for j in range(Select):
    pick = random.choice(PopList)
    CountSmoke = CountSmoke + pick
  CountSmokeTot = CountSmokeTot + CountSmoke
  SmokePlot.append(CountSmokeTot/(Select*(i+1)))

print(CountSmokeTot/(Select*runs))

PlotRuns=[k*Select + Select for k in range(runs)]
# print(PlotRuns)
```

```
0.35237
```

Part 2

without replacement

In [12]:

```
plt.plot(PlotRuns, SmokePlot,'r')
plt.axhline(y= CountSmokeTot/(Select*runs), color='black', linestyle='--')
plt.ylabel('number of trials')
plt.xlabel('fraction of the sample')
plt.title(' Smoking population fraction with replacment')
plt.xscale('log')
plt.show()
```

In [13]:
```
runs = 10000

Select = 10
Population = 1000
NumSmoke = 350

CountSmoke = 0
CountSmokeTot = 0
SmokePlot =[]

for i in range(runs):
  CountSmoke = 0
  for j in range(Select):
    PopList = [*[0 for k in range(Population-NumSmoke)],*[ 1 for k in range(NumSmoke - CountSmoke)]]
    pick = random.choice(PopList)
    CountSmoke = CountSmoke + pick
```

```
  CountSmokeTot = CountSmokeTot + CountSmoke
  SmokePlot.append(CountSmokeTot/(Select*(i+1)))

print(CountSmokeTot/(Select*runs))

PlotRuns=[k*Select + Select for k in range(runs)]
# print(PlotRuns)
```

```
0.34665
```

In [14]:

```
plt.plot(PlotRuns, SmokePlot,'r')
plt.axhline(y= CountSmokeTot/(Select*runs), color='black', linestyle='--')
plt.ylabel('number of trials')
plt.xlabel('fraction of the sample')
plt.title(' Smoking population fraction without replacment')
plt.xscale('log')
plt.show()
```


Smoking population fraction without replacment
0.40
0.38
0.36
0.34
0.32
0.30
number of trials
10^1
10^2
10^3
10^4
10^5
fraction of the sample

In [ ]: