**Assignment 1**

**Simulation of Mining System**

**------------------------------------------------------------------------------------------------------------------------------------------**

**Name:** Ajay Kumar

**Roll:** 12MI31001

**Algorithm :**

* The solution is written in C++ language
* The <random> header is used to generate the random number over the uniform distribution
* The 500 random numbers in between 0 and 1 are generated and stored in array called V.
* Now we have counted for every interval that how many Head and Tail we are getting by the assumption that if V[i] > 0.5 we consider it as head else tail.
* Now for every increasing interval of 10 the number of head and tail we get are stored in array H and T respectively.
* Now Probability is calculated for every trials.

**C++ Implementation**

#include <iostream>

#include <random>

#include <chrono>

#include <vector>

using namespace std;

int main() {

unsigned seed = std::chrono::system\_clock::now().time\_since\_epoch().count();

std::default\_random\_engine generator (seed);

std::uniform\_real\_distribution<double> distribution (0.0,1.0);

int H[50]={0},T[50]={0},interval = 10;

double Values[500];

for (int i=0; i<500;i++){

Values[i] = distribution(generator);

}

for(int i =0;i<500;i+=interval){

for(int j =0;j<i;j++){

if(Values[j]>0.5){

H[j/interval]+=1;

}

else{

T[j/interval] +=1;

}

}

}

cout<<"Head,Tail,P(H),P(T)"<<endl;

for (int i=49; i>=0;i--){

cout<<H[i]<<","<<T[i]<<","<<float(H[i])/float(H[i]+T[i])<<","<<float(T[i])/float(H[i]+T[i])<<endl;

}

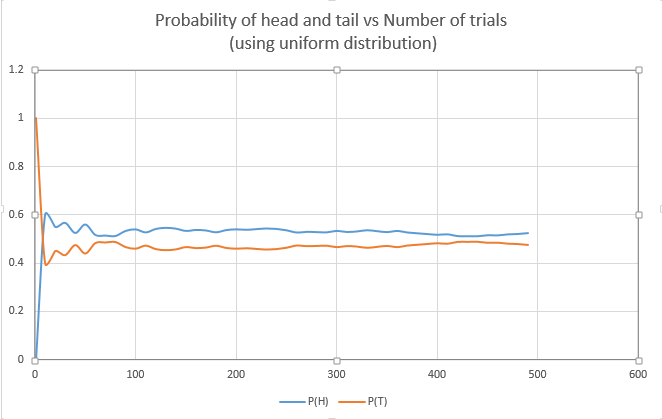
return 0;

}

**Data Generated using uniform distribution for 500 number of trials with coin:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tosses | Head | Tail | P(H) | P(T) |
| 1 | 0 | 1 | 0 | 1 |
| 10 | 6 | 4 | 0.6 | 0.4 |
| 20 | 11 | 9 | 0.55 | 0.45 |
| 30 | 17 | 13 | 0.566667 | 0.433333 |
| 40 | 21 | 19 | 0.525 | 0.475 |
| 50 | 28 | 22 | 0.56 | 0.44 |
| 60 | 31 | 29 | 0.516667 | 0.483333 |
| 70 | 36 | 34 | 0.514286 | 0.485714 |
| 80 | 41 | 39 | 0.5125 | 0.4875 |
| 90 | 48 | 42 | 0.533333 | 0.466667 |
| 100 | 54 | 46 | 0.54 | 0.46 |
| 110 | 58 | 52 | 0.527273 | 0.472727 |
| 120 | 65 | 55 | 0.541667 | 0.458333 |
| 130 | 71 | 59 | 0.546154 | 0.453846 |
| 140 | 76 | 64 | 0.542857 | 0.457143 |
| 150 | 80 | 70 | 0.533333 | 0.466667 |
| 160 | 86 | 74 | 0.5375 | 0.4625 |
| 170 | 91 | 79 | 0.535294 | 0.464706 |
| 180 | 95 | 85 | 0.527778 | 0.472222 |
| 190 | 102 | 88 | 0.536842 | 0.463158 |
| 200 | 108 | 92 | 0.54 | 0.46 |
| 210 | 113 | 97 | 0.538095 | 0.461905 |
| 220 | 119 | 101 | 0.540909 | 0.459091 |
| 230 | 125 | 105 | 0.543478 | 0.456522 |
| 240 | 130 | 110 | 0.541667 | 0.458333 |
| 250 | 134 | 116 | 0.536 | 0.464 |
| 260 | 137 | 123 | 0.526923 | 0.473077 |
| 270 | 143 | 127 | 0.52963 | 0.47037 |
| 280 | 148 | 132 | 0.528571 | 0.471429 |
| 290 | 153 | 137 | 0.527586 | 0.472414 |
| 300 | 160 | 140 | 0.533333 | 0.466667 |
| 310 | 164 | 146 | 0.529032 | 0.470968 |
| 320 | 170 | 150 | 0.53125 | 0.46875 |
| 330 | 177 | 153 | 0.536364 | 0.463636 |
| 340 | 181 | 159 | 0.532353 | 0.467647 |
| 350 | 185 | 165 | 0.528571 | 0.471429 |
| 360 | 192 | 168 | 0.533333 | 0.466667 |
| 370 | 195 | 175 | 0.527027 | 0.472973 |
| 380 | 199 | 181 | 0.523684 | 0.476316 |
| 390 | 203 | 187 | 0.520513 | 0.479487 |
| 400 | 207 | 193 | 0.5175 | 0.4825 |
| 410 | 213 | 197 | 0.519512 | 0.480488 |
| 420 | 215 | 205 | 0.511905 | 0.488095 |
| 430 | 220 | 210 | 0.511628 | 0.488372 |
| 440 | 225 | 215 | 0.511364 | 0.488636 |
| 450 | 232 | 218 | 0.515556 | 0.484444 |
| 460 | 237 | 223 | 0.515217 | 0.484783 |
| 470 | 244 | 226 | 0.519149 | 0.480851 |
| 480 | 250 | 230 | 0.520833 | 0.479167 |
| 490 | 257 | 233 | 0.52449 | 0.47551 |

**The Plot for above result:**

****