**Monte Carlo Simulation**

Monte Carlo simulation is a computerized mathematical technique that allows people to  
account for risk in quantitative analysis and decision making. The technique is used by  
professionals in such widely disparate fields as finance, project management, energy,  
manufacturing, engineering, research and development, insurance, oil & gas,  
transportation, and the environment. Monte Carlo simulation furnishes the decision-maker  
with a range of possible outcomes and the probabilities they will occur for any choice of  
action. It shows the extreme possibilities the outcomes of going for broke and for the most  
conservative decision along with all possible consequences for middle-of-the-road  
decisions.

**Example:**

Dr. Ravi, a dentist schedules all his patients for 30 minute appointments. Some  
of the patients take more or less than 30 minutes depending on the type of dental work to  
be done. The following table shows the summary of the various categories of work, their  
probabilities and the time actually needed to complete the work.

|  |  |  |
| --- | --- | --- |
| Category | Time Required(minutes) | No. of Patients |
| Filling | 45 | 40 |
| Crown | 60 | 15 |
| Cleaning | 15 | 15 |
| Extraction | 45 | 10 |
| Check-up | 15 | 20 |

Simulate the dentist’s clinic for four hours and determine the average waiting time for the  
patients as well as the idleness of the doctor. Assume that all the patients show up at the  
clinic exactly at their scheduled arrival time, starting at 8.00 am. Use the following random  
numbers for handling the above problem: 40, 82, 11, 34, 25, 66, 17, 79

**Code:**

#include<bits/stdc++.h>

using namespace std;

int main()

{

unordered\_map<string,int> serviceTime={{"filling",45},{"crown",60},{"cleaning",15},{"extraction",45},{"check-up",15}};

vector<int> numbers{40,82,11,34,25,66,17,79};

int curTime=0,maxTime=0;

int idle=0;

int wait=0;

for(int i=0;i<8;i++)

{

if(curTime>maxTime)

{

idle+=(curTime-maxTime);

maxTime=curTime;

}

wait+=(maxTime-curTime);

curTime+=30;

string type="";

if(numbers[i]<40)

type="filling";

else if(numbers[i]<55)

type="crown";

else if(numbers[i]<70)

type="cleaning";

else if(numbers[i]<80)

type="extraction";

else

type="check-up";

maxTime+=serviceTime[type];

}

cout<<"Dentist remains idle for "<<idle<<" minutes\n";

cout<<"Average waiting time for patients is :"<<((float)wait/8.0)<<" minutes\n";

return 0;

}

**Output:**

