**Question 1:**

#include<iostream>

#include<string>

#include<fstream>

using namespace std;

void diplay\_menu()

{

string name;

double price;

ifstream infile;

infile.open("menu.txt");

while(!infile.eof())

{

infile>>name>>price;

cout<<name<<price<<endl;

}

infile.close();

}

void order()

{

string dish , x , y , n;

int i=0;

do

{

cout<<"enter the dish you want to order :"<<endl;

cin>>dish;

string a[100];

a[i]=dish;

i++;

cout<<"do you want to order again (y/n)? ";

cin>>x;

}while(x!=y);

}

void generate\_bill(string a[100])

{

int total=0;

string name;

double price;

ifstream infile;

infile.open("menu.txt");

while(!infile.eof())

{

infile>>name>>price;

int i=0;

if(a[i]==name)

{

total=total+price;

}

}

cout<<"your bill is ="<<total;

}

int main()

{

string a[100];

diplay\_menu();

order();

generate\_bill(a);

system("pause");

return 0;

}

**Question 2:**

#include<iostream>

#include<string>

using namespace std;

void matrix(int x[3][3])

{

cout<<"Input the values for a 3x3 matrix :"<<endl;

for(int r=0;r<=2;r++)

{

for(int c=0;c<=2;c++)

{

cin>>x[r][c];

}

}

}

void print\_matrix( int x[3][3])

{

cout<<"matrix is :"<<endl;

for(int r=0;r<=2;r++)

{

for(int c=0;c<=2;c++)

{

cout<<x[r][c]<<"\t";

}

cout<<endl;

}

}

void matrix\_mul(int a[3][3],int b[3][3])

{

cout<<"Input the values for a 3x3 matrix1 :"<<endl;

for(int r=0;r<=2;r++)

{

for(int c=0;c<=2;c++)

{

cin>>a[r][c];

}

}

cout<<"Input the values for a 3x3 matrix2 :"<<endl;

for(int r=0;r<=2;r++)

{

for(int c=0;c<=2;c++)

{

cin>>b[r][c];

}

}

for(int r=0;r<=2;r++)

{

for(int c=0;c<=2;c++)

{

for(int r=0;r<=2;r++)

{

}

}

}

}

void diagonal\_sum(int x[3][3])

{

int sum1=0 , sum2=0;

for(int i=0;i<=2;i++)

{

sum1=sum1+x[i][i];

}

cout<<"left diagonal sum :"<<sum1<<endl;

for(int r=0;r<=2;r++)

{

for(int c=2;c>=0;c--)

{

sum2=sum2+x[r][c];

}

}

cout<<"right diagonal sum :"<<sum2<<endl;

}

void is\_identity(int x[3][3])

{

for(int i=0;i<=2;i++)

{

if(x[i][i]==1)

{}

else

{

cout<<"False , Not an Identity matrix"<<endl;

}

}

for(int r=1;r<=2;r++)

{

if(x[r][0]==0)

{}

else

{

cout<<"False , Not an Identity matrix"<<endl;

}

}

if(x[1][2]==0 && x[2][1]==0)

{}

else

{

cout<<"False , Not an Identity matrix"<<endl;

}

for(int r=1;r<=2;r++)

{

if(x[r][2]==0)

{}

else

{

cout<<"False , Not an Identity matrix"<<endl;

}

}

cout<<"True , an Identity matrix"<<endl;

}

int main()

{

int x[3][3];

int a[3][3] , b[3][3];

matrix(x);

cout<<endl;

print\_matrix(x);

cout<<endl;

matrix\_mul(a,b);

cout<<endl;

diagonal\_sum(x);

cout<<endl;

is\_identity(x);

cout<<endl;

system("pause");

return 0;

}

**Question 3:**

#include<iostream>

#include<string>

using namespace std;

int hotel\_cost(int nights)

{

int hcost;

hcost=140\*nights;

cout<<"hotel cost :";

return hcost;

}

int plane\_ride\_cost(string city)

{

int cost;

if(city=="newyork")

{

cout<<"plane cost ="<<endl;

cost=1000;

return cost;

}

else if(city=="paris")

{

cout<<"plane cost ="<<endl;

cost=2000;

return cost;

}

else

{

cout<<"u entered the wrong city"<<endl;

}

}

int rental\_car\_cost(int days)

{

int rcost;

rcost=40\*days;

cout<<"rental car cost:";

if(days>=7)

{

rcost=rcost-50;

}

else if(days>=3 && days<7)

{

rcost=rcost-20;

}

return rcost;

}

int trip\_cost(string city,int days)

{

int tcost;

cout<<"trip cost";

tcost=rental\_car\_cost(days)+hotel\_cost(days)+plane\_ride\_cost(city);

return tcost;

}

int main()

{

int nights;

cout<<"Enter the no. of nights u want to stay in hotel :";

cin>>nights;

cout<<hotel\_cost(nights);

cout<<endl;

string city;

cout<<"Enter the city u want to go i.e newyork/paris :";

cin>>city;

cout<<plane\_ride\_cost(city);

cout<<endl;

int days;

days=nights;

cout<<rental\_car\_cost(days);

cout<<endl;

cout<<trip\_cost(city,days);

system("pause");

return 0;

}

**Question 4:**

#include<iostream>

#include<string>

#include<fstream>

using namespace std;

void read()

{

int player, hits , walks , outs , atbats;

ifstream infile;

infile.open("baseball.txt");

int h[20][20];

while(!infile.eof())

{

int i=0;

infile>>player>>hits;

h[i][0]=player;

h[i][1]=hits;

i++;

}

int w[20][20];

while(!infile.eof())

{

int i=0;

infile>>player>>walks;

w[i][0]=player;

w[i][1]=walks;

i++;

}

int b[20][20];

while(!infile.eof())

{

int i=0;

infile>>player>>atbats;

b[i][0]=player;

b[i][1]=atbats;

i++;

}

int o[20][20];

while(!infile.eof())

{

int i=0;

infile>>player>>outs;

o[i][0]=player;

o[i][1]=outs;

i++;

}

}

int highesthits(int h[20][20])

{

while(h[20][20])

{

int i=0;

int max=h[0][0];

if(h[i-1][i]>h[0][0])

{

max=h[i-1][i];

}

}

cout<<"max hits =";

return max;

}

int highestwalks(int w[20][20])

{

while(w[20][20])

{

int i=0;

int max=w[0][0];

if(w[i-1][i]>w[0][0])

{

max=w[i-1][i];

}

}

cout<<"max walks =";

return max;

}

int highestouts(int o[20][20])

{

while(o[20][20])

{

int i=0;

int max=o[0][0];

if(o[i-1][i]>o[0][0])

{

max=o[i-1][i];

}

}

cout<<"max outs =";

return max;

}

int highestatbats(int b[20][20])

{

while(b[20][20])

{

int i=0;

int max=b[0][0];

if(b[i-1][i]>b[0][0])

{

max=b[i-1][i];

}

}

cout<<"max walks =";

return max;

}

int main()

{

int h[20][20] , w[20][20] , o[20][20] ,b[20][20];

read();

highesthits(h);

highestwalks(w);

highestouts(o);

highestatbats(b);

system("pause");

return 0;

}

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