# Task 1

#include<iostream>

#include<fstream>

#include<string>

using namespace std;

class dayType {

int dayIndex;

string days[7] = { "Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday" };

public:

dayType()

{}

~dayType()

{}

bool setDay(string day)

{

if (day[0] >= 97 && day[0] <= 122)

day[0] -= 32;

if (day == "Monday" || day == "Tuesday" || day == "Wednesday" || day == "Thursday" || day == "Friday" || day == "Saturday" || day == "Sunday")

{

for (int i = 0; i < 7; i++)

{

if (day == days[i])

{

dayIndex = i;

break;

}

}

return true;

}

return false;

}

string getDay()

{

return days[dayIndex];

}

void printDay()

{

cout << days[dayIndex] << endl;

}

string previousDay()

{

if (dayIndex > 0)

return days[dayIndex - 1];

else if (dayIndex == 0)

return days[6];

}

string nextDay()

{

if (dayIndex < 6)

return days[dayIndex + 1];

else if (dayIndex == 6)

return days[0];

}

string addDays(int number)

{

int n = dayIndex + number;

if (n < 7)

return days[n];

else

{

return days[n % 7];

}

}

};

int main()

{

dayType day;

string dayy;

cout << "Enter day name: ";

cin >> dayy;

if (day.setDay(dayy) == true)

{

cout << "Todays Day:";

day.printDay();

cout << "Previous Day:" << day.previousDay() << endl;

cout << "Next Day:" << day.nextDay() << endl;

int num;

cout << "Enter number of days you want to Add to todays day: "; cin >> num;

cout << "Adding days to todays day:" << day.addDays(num) << endl;

}

else

{

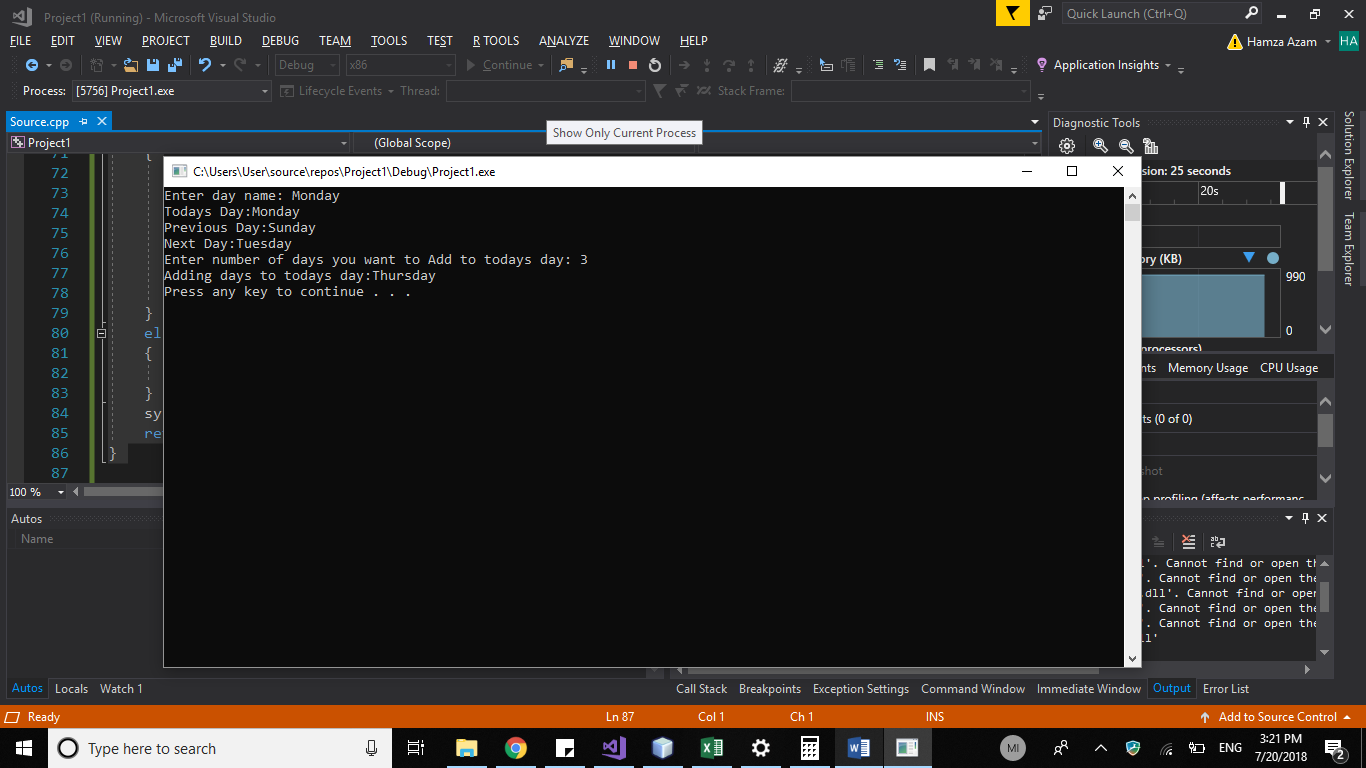
cout << "Incorrect day";

}

system("pause");

return 0;

}



# Task 2

#include<iostream>

#include<fstream>

#include<string>

using namespace std;

class rationalNumber {

int numerator, denominator;

public:

rationalNumber()

{

numerator = denominator = 0;

}

rationalNumber(int n1,int n2)

{

numerator = n1; denominator = n2;

}

~rationalNumber()

{

}

void add(rationalNumber r1, rationalNumber r2)

{

this->numerator = (r1.numerator\*r2.denominator) + (r2.numerator\*r1.denominator);

this->denominator = r1.denominator\*r2.denominator;

int gcd = this->GCD();

this->numerator /= gcd;

this->denominator /= gcd;

}

void sub(rationalNumber r1, rationalNumber r2)

{

this->numerator = (r1.numerator\*r2.denominator) - (r2.numerator\*r1.denominator);

this->denominator = r1.denominator\*r2.denominator;

int gcd = this->GCD();

this->numerator /= gcd;

this->denominator /= gcd;

}

void mul(rationalNumber r1, rationalNumber r2)

{

this->numerator = r1.numerator\*r2.numerator;

this->denominator = r1.denominator\*r2.denominator;

int gcd = this->GCD();

this->numerator /= gcd;

this->denominator /= gcd;

}

void div(rationalNumber r1, rationalNumber r2)

{

this->numerator = r1.numerator\* r2.denominator;

this->denominator = r1.denominator\*r2.numerator;

int gcd = this->GCD();

this->numerator /= gcd;

this->denominator /= gcd;

}

void print()

{

if (this->numerator== this->denominator)

cout << "1\n";

else if (this->numerator==0)

cout << "0\n";

else if (this->denominator == 0)

cout << "Undefined\n";

else

cout << this->numerator << "/" << this->denominator << endl;

}

int GCD()

{

int A = numerator;

int B = denominator;

int Temp;

while (B)

{

Temp = B;

B = A % B;

A = Temp;

}

return A;

}

};

int main()

{

int a, b=0;

while (b == 0)

{

cout << "Enter First Rational number in form a/b (b should not zero): ";

cin >> a; cin.ignore(); cin >> b;

}

rationalNumber r1(a, b);

b = 0;

while (b == 0)

{

cout << "Enter Second Rational number in form a/b (b should not zero): ";

cin >> a; cin.ignore(); cin >> b;

}

rationalNumber r2(a,b);

rationalNumber r3;

r3.add(r1, r2);

cout << "After Adding Both: ";

r3.print();

r3.sub(r1,r2);

cout << "After Subtracting Both: ";

r3.print();

r3.div(r1, r2);

cout << "After Dividing Both: ";

r3.print();

r3.mul(r1, r2);

cout << "After Multiplying Both: ";

r3.print();

system("pause");

return 0;

}

