# **Problems #41 to #45**

===================================

# **Problem 41**

===================

Description: Write A Program To: Weeks and Days

====================================

// Online C++ compiler to run C++ program online

#include <iostream>

using namespace std;

float ReadPositiveNumber(string Message){

int Number = 0;

do{

cout<<Message<<endl;

cin>> Number;

}while(Number < 0 );

return Number;

}

float HoursToDays(float NumberOfHours){

return (float)NumberOfHours / 24;

}

float HoursToWeeks(float NumberOfHours){

return (float)NumberOfHours / 24 / 7;

}

float DaysToWeeks(float NumberOfDays){

return(float)NumberOfDays / 7;

}

int main() {

float NumberOfHours = ReadPositiveNumber("Enter Number OF Hours: ");

float NumberOfDays = HoursToDays(NumberOfHours);

float NumberOfWeeks = DaysToWeeks(NumberOfDays);

cout<<endl;

cout<<"Total Hours = "<<NumberOfHours<<endl;

cout<<"Total Days = "<<NumberOfDays<<endl;

cout<<"Total Weeks = "<< HoursToWeeks(NumberOfHours)<<endl;

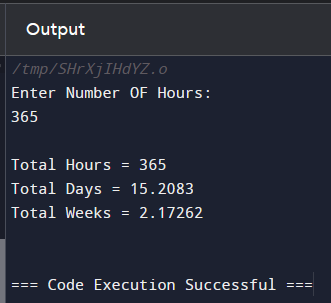
return 0;

}

=====================================================================

The Output:

==========



=====================================================================

# **Problem 42**

===================

Description: Write A Program To Get: Task Duration In Seconds

=================================================

// Online C++ compiler to run C++ program online

#include <iostream>

using namespace std;

struct stTaskDuaration{

int NumberOfDays, NumberOfHours,NumberOfMinutes,

NumberOfSeconds;

};

int ReadPositiveNumber(string Message){

int Number = 0;

do{

cout<<Message<<endl;

cin>> Number;

}while(Number < 0 );

return Number;

}

stTaskDuaration ReadTaskDuration(){

stTaskDuaration TaskDuration;

TaskDuration.NumberOfDays = ReadPositiveNumber(" Enter Number Of Days: ");

TaskDuration.NumberOfHours =ReadPositiveNumber("Enter Number Of Hours: ");

TaskDuration.NumberOfMinutes =ReadPositiveNumber("Enter Number Of Minutes: ");

TaskDuration.NumberOfSeconds =ReadPositiveNumber("Enter Number Of Seconds: ");

return TaskDuration;

}

int TaskDurationInSeconds (stTaskDuaration TaskDuration){

int DurationInSeconds = 0;

DurationInSeconds = TaskDuration.NumberOfDays \* 24

\* 60 \* 60;

DurationInSeconds += TaskDuration.NumberOfHours \* 60

\* 60;

DurationInSeconds += TaskDuration.NumberOfMinutes \* 60;

DurationInSeconds += TaskDuration.NumberOfSeconds;

return DurationInSeconds;

}

int main() {

cout<<"\nTask Duration In Seconds: \n"<<TaskDurationInSeconds(ReadTaskDuration());

cout<<endl;

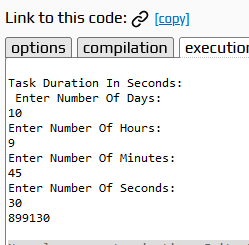
return 0;

}

=====================================================================

The Output:

===========



=====================================================================

# **Problem 43**

======================

Description:Write A Program For: Seconds To Days Hours Minutes Seconds

=====================================================================

// Online C++ compiler to run C++ program online

#include <iostream>

#include <cmath>

using namespace std;

struct stTaskDuaration{

int NumberOfDays, NumberOfHours,NumberOfMinutes,

NumberOfSeconds;

};

int ReadPositiveNumber(string Message){

int Number = 0;

do{

cout<<Message<<endl;

cin>> Number;

}while(Number < 0 );

return Number;

}

stTaskDuaration SecondsToTaskDuration(int TotalSeconds){

stTaskDuaration TaskDuration;

const int SecondsPerDay = 24 \* 60 \* 60;

const int SecondsPerHour = 60 \* 60;

const int SecondPerMinute = 60;

int Remainder = 0;

TaskDuration.NumberOfDays = floor(TotalSeconds / SecondsPerDay);

Remainder = TotalSeconds % SecondsPerDay;

TaskDuration.NumberOfHours = floor(Remainder / SecondsPerHour);

Remainder = TotalSeconds % SecondsPerHour;

TaskDuration.NumberOfMinutes = floor(Remainder / SecondPerMinute);

Remainder = TotalSeconds % SecondPerMinute;

TaskDuration.NumberOfSeconds = Remainder;

return TaskDuration;

}

void PrintTaskDurationDetails(stTaskDuaration TaskDuration){

cout<<"\n";

cout<<TaskDuration.NumberOfDays<<":"

<<TaskDuration.NumberOfHours<<":"

<<TaskDuration.NumberOfMinutes<<":"

<<TaskDuration.NumberOfSeconds<<"\n";

}

int main() {

int TotalSeconds = ReadPositiveNumber("Please Enter Total Seconds: ");

PrintTaskDurationDetails(SecondsToTaskDuration(TotalSeconds));

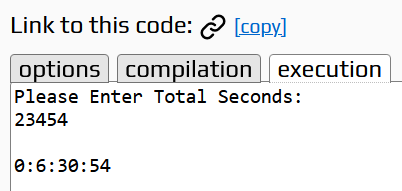
return 0;

}

=====================================================================

The Output:

==========



=====================================================================

# **Problem 44**

==================

Description: Write A Program: Day Of Week

===============================

#include <iostream>

using namespace std;

enum enDayOfWeek {Sat = 1, Sun = 2, Mon = 3, Tue = 4, Wed = 5 , Thu = 6, Fri= 7};

int ReadNumberInRange(string Message,int From, int To){

int Number = 0;

do{

cout<<Message<<endl;

cin>> Number;

}while(Number < From || Number > To);

return Number;

}

enDayOfWeek ReadDayOfWeek(){

return (enDayOfWeek)ReadNumberInRange("Please Enter Number Of day: ",1,7);

}

string GetDayOfweek(enDayOfWeek Day){

switch(Day){

case enDayOfWeek::Sat:

return "Saturday";

case enDayOfWeek::Sun:

return "Sunday";

case enDayOfWeek::Mon:

return "Monday";

case enDayOfWeek::Tue:

return "Tuesday";

case enDayOfWeek::Wed:

return "Wednesday";

case enDayOfWeek::Thu:

return "Thursday";

case enDayOfWeek::Fri:

return "Friday";

default:

return "None A Valid Day";

}

}

int main(){

cout<<GetDayOfweek(ReadDayOfWeek())<<endl;

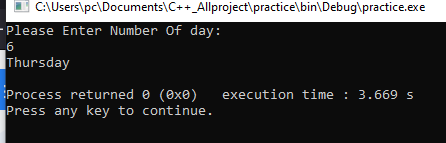
return 0;

}

=====================================================================

The OutPut:

=========



=====================================================================

# **Problem 45**

========================

Description: Write A Program To get Month Of The Year

=========================================

#include <iostream>

using namespace std;

enum enMonthOfYear {Jan = 1, Feb = 2, Mar = 3, Apr = 4, May = 5 , Jun = 6, Jul= 7, Aug = 8, Sep = 9, Oct = 10, Nov = 11, Dec = 12};

int ReadNumberInRange(string Message,int From, int To){

int Number = 0;

do{

cout<<Message<<endl;

cin>> Number;

}while(Number < From || Number > To);

return Number;

}

enMonthOfYear ReadMonthOfYear(){

return (enMonthOfYear)ReadNumberInRange("Please Enter Number of Month from [1, 12]: ",1,12);

}

string GetMonthOfYear(enMonthOfYear Month){

switch(Month){

case enMonthOfYear::Jan:

return "January";

case enMonthOfYear::Feb:

return "February";

case enMonthOfYear::Mar:

return "March";

case enMonthOfYear::Apr:

return "April";

case enMonthOfYear::May:

return "May";

case enMonthOfYear::Jun:

return "Junuary";

case enMonthOfYear::Jul:

return "Julay";

case enMonthOfYear::Aug:

return "August";

case enMonthOfYear::Sep:

return "September";

case enMonthOfYear::Oct:

return "October";

case enMonthOfYear::Nov:

return "November";

case enMonthOfYear::Dec:

return "December";

default:

return "None A Valid Month";

}

}

int main(){

cout<<GetMonthOfYear(ReadMonthOfYear())<<endl;

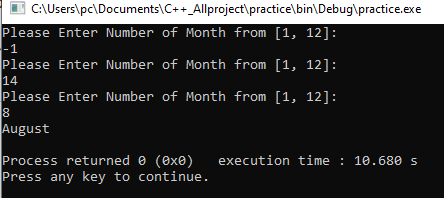
return 0;

}

=====================================================================

The Output

========



=====================================================================