8 - Algorithms & Problem Solving Level 4
Mohammed Abu-Hadhoud

Problem #01 - Number To Text

Write a program to read a number and print the text of that number.

اكتب برنامجًا لقراءة رقم وطباعة نص هذا الرقم.

```
Enter a Number : 5843

Five Thousand, Eight Hundred Forty-Three
```

```
Enter a Number : 8476362741

Eight Billion, Four Hundred Seventy-Six Million, Three Hundred Sixty-Two Thousand, Seven Hundred Forty-One
```

Solution:

```
#include<iostream>

using namespace std;

//long long int ReadNumber()
int ReadNumber()
{
	//long long int Num;
	int Num;
	cout << "\n\tEnter a Number : ";
	cin >> Num;
	return Num;
}

//string NumberToText(long long int Num)
string NumberToText(int Num)
{
	string Arr1To19[] = {"", "One", "Two", "Three", "Four", "Five", "Six",
		"Seven", "Eight", "Nine", "Ten", "Eleven", "Twelve", "Thirteen",
		"Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen",
		"Nineteen" };

	string Arr20To99[] = {"", "", "Twenty", "Thirty", "Forty", "Fifty",
		"Sixty", "Seventy", "Eighty", "Ninety" };

	if (Num == 0)
	{
		return "";
	}

	if (Num >= 1 && Num <= 19)
	{
		return Arr1To19[Num];
	}

	if (Num >= 20 && Num <= 99)
	{
		return Arr20To99[Num / 10] + "-" + NumberToText(Num % 10);
	}

	if (Num >= 100 && Num <= 999)
	{
		return NumberToText(Num / 100) + " Hundred " + NumberToText(Num % 100);
	}
```

```
	if (Num >= 1000 && Num <= 999999)
	{
		return NumberToText(Num / 1000) + " Thousand, " + NumberToText(Num % 1000);
	}

	if (Num >= 1000000 && Num <= 999999999)
	{
		return NumberToText(Num / 1000000) + " Million, " + NumberToText(Num % 1000000);
	}

	/*if (Num >= 1000000000 && Num <= 999999999999)
	{
		return NumberToText(Num / 1000000000) + " Billion, " + NumberToText(Num % 1000000000);
	}

	else
	{
		return NumberToText(Num / 1000000000000) + " Trillion, " + NumberToText(Num % 1000000000000);
	}*/
}

int main()
{
	system("color f0");

	//long long int Num = ReadNumber();

	int Num = ReadNumber();
	cout << "\n\t" << NumberToText(Num) << endl;

	system("pause>0");
}
```

==========

Problem #02 - Leap Year:

Write a program to check if year is a Leap year or not.

اكتب برنامجًا لمعرفة ما إذا كانت السنة سنة كبيسة أم لا.

Note: All years which are perfectly divisible by 4 are Leap years except for century years (years ending with 00), which are leap years only if they are perfectly divisible by 400.

ملحوظة:

- **إذا كانت السنة قابلة للقسمة على ٤، فهي سنة كبيسة.**
- **ولكن إذا كانت السنة قابلة للقسمة على ١٠٠، فهي ليست سنة كبيسة.**
- **إلا إذا كانت السنة أيضًا قابلة للقسمة على ٤٠٠، فهي سنة كبيسة.**
- **على سبيل المثال، عام ٢٠٠٠ سنة كبيسة، بينما عام ١٩٠٠ ليست سنة كبيسة.**

```
Enter a year to check : 1968

YES, This year (1968) is a leap year.
```

```
Enter a year to check : 1971

NO, This year (1971) is Not a leap year.
```

Solution:

```
#include<iostream>
using namespace std;

short ReadYear()
{
	short Year;
	cout << "\n\tEnter a year to check : ";
	cin >> Year;
	return Year;
}

bool IsLeapYear(short Year)
{
	// leap year if perfectly divisible by 400
	// السنة الكبيسة إذا كانت قابلة للقسمة على 400
	if (Year % 400 == 0)
	{
		return true;
	}

	// not a leap year if divisible by 100 - but not divisible by 400
	// ولكن لا يقبل القسمة على 400 - ليست سنة كبيسة إذا كانت قابلة للقسمة على 100
	else if (Year % 100 == 0)
	{
		return false;
	}

	// leap year if not divisible by 100 - but divisible by 4
	// لكن قابل للقسمة على 4 - السنة الكبيسة إذا لم تكن قابلة للقسمة على 100
	else if (Year % 4 == 0)
	{
		return true;
	}

	// all other years are not leap years - جميع السنوات الأخرى ليست سنوات كبيسة
	else
	{
		return false;
	}
}

void PrintResult()
{
	short Year = ReadYear();
	if (IsLeapYear(Year))
		cout << "\n\tYES, This year (" << Year << ") is a leap year." << endl;
	else
		cout << "\n\tNO, This year (" << Year << ") is Not a leap year." << endl;
}

int main()
{
	system("color f0");
	PrintResult();
	system("pause>0");
}
```

Problem #03: Leap Year (One Line Of Code) – (Optimization : تحسين)

Solve leap year problem with one line of code only.

Note: Use Logical Operators.

حل مشكلة السنة الكبيسة بسطر واحد فقط من التعليمات البرمجية.

ملاحظة: استخدم العوامل المنطقية.

```
Enter a year to check : 1968

YES, This year (1968) is a leap year.
```

```
Enter a year to check : 1971

NO, This year (1971) is Not a leap year.
```

Solution:

```
#include<iostream>

using namespace std;

short ReadYear()
{
	short Year;
	cout << "\n\tEnter a year to check : ";
	cin >> Year;
	return Year;
}

bool IsLeapYear(short Year)
{
	return (Year % 4 == 0 && Year % 100 != 0) || Year % 400 == 0 ? true : false;
}

void PrintResult()
{
	short Year = ReadYear();

	if (IsLeapYear(Year))
		cout << "\n\tYES, This year (" << Year << ") is a leap year." << endl;
	else
		cout << "\n\tNO, This year (" << Year << ") is Not a leap year." << endl;
}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
}
```

Problem #04: Number Of Days-Hours-Minutes-Seconds in a Year

Write a program to print Number of: Days - Hours - Minutes - Seconds in a certain year.

اكتب برنامج لطباعة عدد: الأيام - الساعات - الدقائق - الثوانى فى سنة معينة.

```
    Enter a year to check : 2000

    Number of Days    in Year (2000) is: 366
    Number of Hours   in Year (2000) is: 8784
    Number of Minutes in Year (2000) is: 527040
    Number of Seconds in Year (2000) is: 31622400
```

```
    Enter a year to check : 1971

    Number of Days    in Year (1971) is: 365
    Number of Hours   in Year (1971) is: 8760
    Number of Minutes in Year (1971) is: 525600
    Number of Seconds in Year (1971) is: 31536000
```

Solution:

```
#include<iostream>

using namespace std;

short ReadYear()
{
      short Year;
      cout << "\n\tEnter a year to check : ";
      cin >> Year;
      return Year;
}

bool IsLeapYear(short Year)
{
      return (Year % 4 == 0 && Year % 100 != 0) || Year % 400 == 0;
}

short NumbersOfDaysInYear(short Year)
{
      return IsLeapYear(Year) ? 366 : 365;
}

short NumbersOfHoursInYear(short Year)
{
      return NumbersOfDaysInYear(Year) * 24;
}

int NumbersOfMinutsInYear(short Year)
{
      return NumbersOfHoursInYear(Year) * 60;
}

int NumbersOfSecondsInYear(short Year)
{
      return NumbersOfMinutsInYear(Year) * 60;
}
```

```
void PrintResult()
{
	short Year = ReadYear();
	cout << "\n\tNumber of Days    in Year (" << Year << ") is: " <<
	NumbersOfDaysInYear(Year) << endl;
	cout << "\tNumber of Hours   in Year (" << Year << ") is: " <<
	NumbersOfHoursInYear(Year) << endl;
	cout << "\tNumber of Minutes in Year (" << Year << ") is: " <<
	NumbersOfMinutsInYear(Year) << endl;
	cout << "\tNumber of Seconds in Year (" << Year << ") is: " <<
	NumbersOfSecondsInYear(Year) << endl;
}
int main()
{
	system("color f0");
	PrintResult();
	system("pause>0");
}
```

==========

Problem #05: Number Of Days-Hours-Minutes-Seconds in a Month

Write a program to print Number of: Days - Hours - Minutes - Seconds in a certain Month.

اكتب برنامج لطباعة عدد: الأيام - الساعات - الدقائق - الثواني في شهر معين.

```
Enter a year to check : 2024

Enter a Month to check : 2

Number of Days    in Month (2) is: 29
Number of Hours   in Month (2) is: 696
Number of Minutes in Month (2) is: 41760
Number of Seconds in Month (2) is: 2505600
```

```
Enter a year to check : 2023

Enter a Month to check : 2

Number of Days    in Month (2) is: 28
Number of Hours   in Month (2) is: 672
Number of Minutes in Month (2) is: 40320
Number of Seconds in Month (2) is: 2419200
```

```
Enter a year to check : 2024

Enter a Month to check : 5

Number of Days    in Month (5) is: 31
Number of Hours   in Month (5) is: 744
Number of Minutes in Month (5) is: 44640
Number of Seconds in Month (5) is: 2678400
```

```
Enter a year to check : 2024

Enter a Month to check : 6

Number of Days    in Month (6) is: 30
Number of Hours   in Month (6) is: 720
Number of Minutes in Month (6) is: 43200
Number of Seconds in Month (6) is: 2592000
```

Solution:

```
#include<iostream>
using namespace std;

short ReadYear()
{
	short Year;
	cout << "\n\tEnter a year to check : ";
	cin >> Year;
	return Year;
}
short ReadMonth()
{
	short Month;
	cout << "\n\tEnter a Month to check : ";
	cin >> Month;
	return Month;
}
```

```
bool IsLeapYear(short Year)
{
     return (Year % 4 == 0 && Year % 100 != 0) || Year % 400 == 0;
}

short NumbersOfDaysInMonth(short Year, short Month)
{
     if (Month < 1 || Month > 12)
     {
          return 0;
     }

     if (Month == 2)
     {
          return IsLeapYear(Year) ? 29 : 28;
     }

     short Arr31Deys[7] = { 1, 3, 5, 7, 8, 10, 12 };
     for (short M = 1; M <= 7; M++)
     {
          if (Arr31Deys[M-1] == Month)
               return 31;
     }
     // If you reach here then its 30 days
     return 30;
}

short NumbersOfHoursInMonth(short Year, short Month)
{ return NumbersOfDaysInMonth(Year, Month) * 24; }

int NumbersOfMinutsInMonth(short Year, short Month)
{ return NumbersOfHoursInMonth(Year, Month) * 60; }

int NumbersOfSecondsInMonth(short Year, short Month)
{ return NumbersOfMinutsInMonth(Year, Month) * 60; }

void PrintResult()
{
     short Year = ReadYear();
     short Month = ReadMonth();
     cout << "\n\tNumber of Days    in Month (" << Month << ") is: " <<
     NumbersOfDaysInMonth(Year, Month) << endl;
     cout << "\tNumber of Hours   in Month (" << Month << ") is: " <<
     NumbersOfHoursInMonth(Year, Month) << endl;
     cout << "\tNumber of Minutes in Month (" << Month << ") is: " <<
     NumbersOfMinutsInMonth(Year, Month) << endl;
     cout << "\tNumber of Seconds in Month (" << Month << ") is: " <<
     NumbersOfSecondsInMonth(Year, Month) << endl;
}

int main()
{
     system("color f0");
     PrintResult();
     system("pause>0");
}
```

Problem #06: Number Of Days In a Month Short Logic

Write a program to print Number of Days in a certain Month.

اكتب برنامج لطباعة عدد الأيام في شهر معين (حل مختصر للمشكلة السابقة).

Note: Two lines of code (Logic) / ملاحظة: سطرين من التعليمات البرمجية (المنطق)

```
Enter a year to check : 2024

Enter a Month to check : 2

Number of Days in Month (2) is: 29
```

```
Enter a year to check : 2023

Enter a Month to check : 2

Number of Days in Month (2) is: 28
```

```
Enter a year to check : 2024

Enter a Month to check : 5

Number of Days in Month (5) is: 31
```

```
Enter a year to check : 2024

Enter a Month to check : 6

Number of Days in Month (6) is: 30
```

Solution:

```
#include<iostream>

using namespace std;

short ReadYear()
{
	short Year;
	cout << "\n\tEnter a year to check : ";
	cin >> Year;
	return Year;
}

short ReadMonth()
{
	short Month;
	cout << "\n\tEnter a Month to check : ";
	cin >> Month;
	return Month;
}


bool IsLeapYear(short Year)
{
	return (Year % 4 == 0 && Year % 100 != 0) || Year % 400 == 0;
}


short NumbersOfDaysInMonth(short Year, short Month)
{
	if (Month < 1 || Month > 12)
	{
		return 0;
	}

	short DeysInMonth[12] = {31,28,31,30,31,30,31,31,30,31,30,31};

	return (Month == 2) ? (IsLeapYear(Year) ? 29 : 28) : DeysInMonth[Month - 1];
}
```

```
void PrintResult()
{
	short Year = ReadYear();
	short Month = ReadMonth();

	cout << "\n\tNumber of Days in Month (" << Month << ") is: " <<
	NumbersOfDaysInMonth(Year, Month) << endl;
}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
}
```

==========

Problem #07: Day Name

Write a program to read a date, and print the day name of week.

كتابة برنامج لقراءة التاريخ، وطباعة اسم اليوم من الأسبوع. (باستخدام هذا القانون)

$$a = \left\lfloor \frac{14 - month}{12} \right\rfloor$$

$$y = year - a$$

$$m = month + 12a - 2$$

For the Julian calendar:

$$d = \left(5 + day + y + \left\lfloor \frac{y}{4} \right\rfloor + \left\lfloor \frac{31m}{12} \right\rfloor\right) \bmod 7$$

For the Gregorian calendar:

$$d = \left(day + y + \left\lfloor \frac{y}{4} \right\rfloor - \left\lfloor \frac{y}{100} \right\rfloor + \left\lfloor \frac{y}{400} \right\rfloor + \left\lfloor \frac{31m}{12} \right\rfloor\right) \bmod 7$$

The value of d is 0 for a Sunday, 1 for a Monday, 2 for a Tuesday etc.

```
Please enter a year : 2022

Please enter a month: 9

Please enter a day  : 20

Date      : 20/9/2022
Day Order : 2
Day Name  : Tue
```

Solution:

```
#include<iostream>

using namespace std;

short ReadYear()
{
	short Y;
	cout << "\n\tPlease enter a year : ";
	cin >> Y;
	return Y;
}

short ReadMonth()
{
	short M;
	cout << "\n\tPlease enter a month: ";
	cin >> M;
	return M;
}
```

```cpp
short ReadDay()
{
	short D;
	cout << "\n\tPlease enter a day  : ";
	cin >> D;
	return D;
}

short DayNameOfWeek(short Year, short Month, short Day)
{
	short a = ((14 - Month) / 12);

	short y = Year - a;

	short m = Month + (12 * a) - 2;

	// For the Greforian Calendar:

	short d = (Day + y + (y / 4) - (y / 100) + (y / 400) + ((31 * m) / 12)) % 7;

	return d;
	// The value of d is 0 for a Sunday, 1 for a Monday, 2 for a Tuesday etc..
}

string DayName(short DN)
{
	string DayName[7] = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };
	return DayName[DN];
}

void PrintResult()
{
	short Y = ReadYear();
	short M = ReadMonth();
	short D = ReadDay();

	cout << "\n\tDate      : " << D << "/" << M << "/" << Y ;
	cout << "\n\tDay Order : " << DayNameOfWeek(Y,M,D) ;
	cout << "\n\tDay Name  : " << DayName(DayNameOfWeek(Y, M, D)) << endl;
}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
}
```

Problem #08: Month Calendar

Write a program to print Month Calendar.

اكتب برنامج لطباعة التقويم الشهري.

```
    PLease enter a year : 2024
    Please Enter a month (1 to 12): 5

    _________________May_________________

      Sun   Mon   Tue   Wed   Thu   Fri   Sat
                        1     2     3     4
        5     6     7     8     9    10    11
       12    13    14    15    16    17    18
       19    20    21    22    23    24    25
       26    27    28    29    30    31

    ______________________________________
```

Solution:

```
#include<iostream>
#include<iomanip>
using namespace std;

short ReadYear()
{
      short y;
      cout << "\tPLease enter a year : ";
      cin >> y;
      return y;
}

short ReadMonth()
{
      short m;
      do
      {
            cout << "\tPlease Enter a month (1 to 12): ";
            cin >> m;
      } while (m < 1 || m > 12);
      return m;
}

bool IsLeapYear(short year)
{
      return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true :
false;
}

short DayNumberInWeek(short Year, short Month, short Day)
{
      short a = ((14 - Month) / 12);
      short y = Year - a;
      short m = Month + (12 * a) - 2;
      // For the Greforian Calendar:
      short d = (Day + y + (y / 4) - (y / 100) + (y / 400) + ((31 * m) / 12)) % 7;
      return d;
}
```

```
short NumberOfDaysInMonth(short Year, short Month)
{
	if (Month < 1 || Month > 12)
	{
		return 0;
	}
	short NumberOfDays[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

	return (Month == 2) ? (IsLeapYear(Year) ? 29 : 28) : NumberOfDays[Month - 1];
}

string ShortNameForMonth(short Month)
{
	string ShortName[] = { "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
	"Aug", "Sept", "Oct", "Nov", "Dec" };
	return ShortName[Month - 1];
}

string ShortNameForDay(short Day)
{
	string ShortName[] = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };
	return ShortName[Day - 1];
}

void PrintMonthCalendar()
{
	short y = ReadYear();
	short m = ReadMonth();
	short ND = NumberOfDaysInMonth(y, m);
	short current = DayNumberInWeek(y, m, 1);
	string MN = ShortNameForMonth(m);
	cout << "\n\t_________________";
	cout << MN;
	cout << "_________________\n\n";
	cout << "\t";
	for (short i = 1; i <= 7; i++)
	{
		cout << setw(5) << ShortNameForDay(i);
	}
	cout << endl;
	short k = 0;
	cout << "\t";
	for (k = 0; k < current; k++)
	{
		cout <<  "     ";
	}
	for (short j = 1; j <= ND; j++)
	{
		cout << setw(5) << j;
		if (++k > 6)
		{
			k = 0;
			cout << endl;
			cout << "\t";
		}
	}
	cout << "\n\t_____________________________________\n\n";
}
```

```
int main()
{
	system("color f0");

	PrintMonthCalendar();

	system("pause>0");
	return 0;
}
```

==========

Problem #09: Year Calendar

Write a program to print Year Calendar.

اكتب برنامج لطباعة التقويم السنوي.

```
PLease enter a year : 2024

====================================
          Calendar - 2024
====================================

                Jan

 Sun  Mon  Tue  Wed  Thu  Fri  Sat
                1    2    3    4    5    6
   7    8    9   10   11   12   13
  14   15   16   17   18   19   20
  21   22   23   24   25   26   27
  28   29   30   31

                Feb

 Sun  Mon  Tue  Wed  Thu  Fri  Sat
                               1    2    3
   4    5    6    7    8    9   10
  11   12   13   14   15   16   17
  18   19   20   21   22   23   24
  25   26   27   28   29
```

```
                Mar

 Sun  Mon  Tue  Wed  Thu  Fri  Sat
                               1    2
   3    4    5    6    7    8    9
  10   11   12   13   14   15   16
  17   18   19   20   21   22   23
  24   25   26   27   28   29   30
  31

                Apr

 Sun  Mon  Tue  Wed  Thu  Fri  Sat
                1    2    3    4    5    6
   7    8    9   10   11   12   13
  14   15   16   17   18   19   20
  21   22   23   24   25   26   27
  28   29   30
```

Solving:

```
#include<iostream>
#include<iomanip>
using namespace std;

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}
```

```
short DayNumberInWeek(short Year, short Month, short Day)
{
	short a = ((14 - Month) / 12);
	short y = Year - a;
	short m = Month + (12 * a) - 2;
	// For the Greforian Calendar:
	short d = (Day + y + (y / 4) - (y / 100) + (y / 400) + ((31 * m) / 12)) % 7;
	return d;
}

short NumberOfDaysInMonth(short Year, short Month)
{
	if (Month < 1 || Month > 12)
	{
		return 0;
	}

	short NumberOfDays[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

	return (Month == 2) ? (IsLeapYear(Year) ? 29 : 28) : NumberOfDays[Month - 1];
}

string ShortNameForMonth(short Month)
{
	string ShortName[] = { "Jan", "Feb", "Mar", "Apr", "May", "Jun",
				"Jul", "Aug", "Sept", "Oct", "Nov", "Dec" };

	return ShortName[Month - 1];
}

string ShortNameForDay(short Day)
{
	string ShortName[] = { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };

	return ShortName[Day - 1];
}

void PrintMonthCalendar(short y, short m)
{
	short ND = NumberOfDaysInMonth(y, m);

	short current = DayNumberInWeek(y, m, 1);
	string MN = ShortNameForMonth(m);

	cout << "\n\t_________________";
	cout << MN;
	cout << "_________________\n\n";

	cout << "\t";
	for (short i = 1; i <= 7; i++)
	{
		cout << setw(5) << ShortNameForDay(i);
	}
	cout << endl;

	short k = 0;

	cout << "\t";
```

```
	for (k = 0; k < current; k++)
	{
		cout <<  "     ";
	}


	for (short j = 1; j <= ND; j++)
	{
		cout << setw(5) << j;

		if (++k > 6)
		{
			k = 0;
			cout << endl;
			cout << "\t";
		}
	}
	cout << "\n\t_____________________________________\n\n";
}

void PrintCalendarYear()
{
	short y = ReadYear();

	cout << "\n\t=====================================\n";
	cout << "\t           Calendar - " << y;
	cout << "\n\t=====================================\n";

	for (short i = 1; i <= 12; i++)
	{
		PrintMonthCalendar(y, i);
	}
}

int main()
{
	system("color f0");

	PrintCalendarYear();

	system("pause>0");
	return 0;
}
```

Problem #10: Days From The Beginning Of Year

Write a program to print the total days from the beginning of the year.

اكتب برنامجًا لطباعة إجمالي الأيام من بداية العام.

```
    PLease enter a year : 2022
    Please Enter a month (1 to 12): 9
    Please Enter a day (1 to 31): 20

    Number of Days from the begining of the year is: 263
```

Solving:

```
#include<iostream>
#include<iomanip>
using namespace std;

short ReadYear()
{     short y;
      cout << "\tPLease enter a year : ";
      cin >> y;
      return y;
}

short ReadMonth()
{     short m;
      do
      {     cout << "\tPlease Enter a month (1 to 12): ";
            cin >> m;
      } while (m < 1 || m > 12);
      return m;
}

short ReadDay()
{     short d;
      do
      {     cout << "\tPlease Enter a day (1 to 31): ";
            cin >> d;
      } while (d < 1 || d > 31);
      return d;
}

bool IsLeapYear(short year)
{ return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false; }

short NumberOfDaysInMonth(short Year, short Month)
{
      if (Month < 1 || Month > 12)
      {
            return 0;
      }

      short NumberOfDays[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

      return (Month == 2) ? (IsLeapYear(Year) ? 29 : 28) : NumberOfDays[Month - 1];

}
```

```
short NumberOfDaysFromTheBeginingOfTheYear(short Year, short Month, short Day)
{     short TotalDays = 0;
      for(short i = 1; i <= Month-1; i++)
      {
            TotalDays += NumberOfDaysInMonth(Year, i);
      }
      TotalDays += Day;
      return TotalDays;
}

void PrintResult()
{
      short Year = ReadYear();
      short Month = ReadMonth();
      short Day = ReadDay();
      cout << "\n\tNumber of Days from the begining of the year is: "
            << NumberOfDaysFromTheBeginingOfTheYear(Year, Month, Day) << endl;
}

int main()
{
      system("color f0");
      PrintResult();

      system("pause>0");
      return 0;
}
```

==========

Problem #11: Date From Day Order In a Year

Write a program to print total days from the beginning of year, then Take the total days and convert them back to date.

اكتب برنامجًا لطباعة إجمالي الأيام من بداية العام، ثم خذ إجمالي الأيام وقم بتحويلها إلى تاريخ.

```
PLease enter a year : 2022
Please Enter a month (1 to 12): 9
Please Enter a day (1 to 31): 20

Number Of Days From The Begining Of The Year : 263

Date for (263) is : 20/9/2022
```

Solving:

```
#include<iostream>
#include<iomanip>

using namespace std;

short ReadYear()
{
      short y;
      cout << "\tPLease enter a year : ";
      cin >> y;
      return y;
}
```

```
short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month (1 to 12): ";
		cin >> m;
	} while (m < 1 || m > 12);
	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day (1 to 31): ";
		cin >> d;
	} while (d < 1 || d > 31);
	return d;
}

bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

short NumberOfDaysInMonth(short Year, short Month)
{
	if (Month < 1 || Month > 12)
	{
		return 0;
	}

	short NumberOfDays[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
	return (Month == 2) ? (IsLeapYear(Year) ? 29 : 28) : NumberOfDays[Month - 1];
}

short NumberOfDaysFromTheBeginingOfTheYear(short Year, short Month, short Day)
{
	short TotalDays = 0;

	for (short i = 1; i <= Month - 1; i++)
	{
		TotalDays += NumberOfDaysInMonth(Year, i);
	}

	TotalDays += Day;

	return TotalDays;
}

struct stDate {
	short Year;
	short Month;
	short Day;
};
```

```
stDate GetDateFromDayOrderInYear(short Year, short DateOrderInYear)
{
	stDate Date;

	short RemainingDays = DateOrderInYear;
	short MonthDays = 0;

	Date.Year = Year;
	Date.Month = 1;

	while (true)
	{
		MonthDays = NumberOfDaysInMonth(Year, Date.Month);

		if (RemainingDays > MonthDays)
		{
			RemainingDays = RemainingDays - MonthDays;
			Date.Month++;
		}
		else
		{
			Date.Day = RemainingDays;
			break;
		}
	}
	return Date;
}

void PrintResult()
{
	short Year = ReadYear();
	short Month = ReadMonth();
	short Day = ReadDay();

	short DaysOrderInYear = NumberOfDaysFromTheBeginingOfTheYear(Year, Month, Day);

	stDate Date = GetDateFromDayOrderInYear(Year, DaysOrderInYear);


	cout << "\n\tNumber Of Days From The Begining Of The Year : "
	     << DaysOrderInYear << endl;

	cout << "\n\tDate for (" << DaysOrderInYear << ") is : ";

	cout << Date.Day << "/" << Date.Month << "/" << Date.Year << endl;
}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
}
```

Problem #12: Add Days To Date

Write a program to read date and read how many days to add to it, print the result on screen.
كتابة برنامج لقراءة التاريخ وقراءة عدد الأيام التى يجب إضافتها إليه، وطباعة النتيجة (التاريخ الجديد) على الشاشة.

```
    Please Enter a day (1 to 31): 10
    Please Enter a month (1 to 12): 10
    PLease enter a year : 2022

    How many days to add: 2500

    Date after adding (2500) Days is : 14/8/2029
```

Solving:

```
#include<iostream>
#include<iomanip>

using namespace std;

struct stDate {
      short Year;
      short Month;
      short Day;
};

short ReadYear()
{
      short y;
      cout << "\tPLease enter a year : ";
      cin >> y;
      return y;
}

short ReadMonth()
{
      short m;
      do
      {
            cout << "\tPlease Enter a month (1 to 12): ";
            cin >> m;
      } while (m < 1 || m > 12);

      return m;
}

short ReadDay()
{
      short d;
      do
      {
            cout << "\tPlease Enter a day (1 to 31): ";
            cin >> d;
      } while (d < 1 || d > 31);

      return d;
}
```

```
stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

short ReadDaysToAdd()
{
	short Days;
	cout << "\n\tHow many days to add: ";
	cin >> Days;
	return Days;
}

bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

short NumberOfDaysInMonth(short Year, short Month)
{
	if (Month < 1 || Month > 12)
	{
		return 0;
	}

	short NumberOfDays[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

	return (Month == 2) ? (IsLeapYear(Year) ? 29 : 28) : NumberOfDays[Month - 1];
}

short NumberOfDaysFromTheBeginingOfTheYear(short Year, short Month, short Day)
{
	short TotalDays = 0;

	for (short i = 1; i <= Month - 1; i++)
	{
		TotalDays += NumberOfDaysInMonth(Year, i);
	}

	TotalDays += Day;

	return TotalDays;
}

stDate DateAddDays(stDate Date, short DateOrderInYear)
{
	short RemainingDays = DateOrderInYear;

	short MonthDays = 0;

	Date.Month = 1;
```

```
	while (true)
	{
		MonthDays = NumberOfDaysInMonth(Date.Year, Date.Month);

		if (RemainingDays > MonthDays)
		{
			RemainingDays -= MonthDays;
			Date.Month++;

			if (Date.Month > 12)
			{
				Date.Month = 1;
				Date.Year++;
			}
		}
		else
		{
			Date.Day = RemainingDays;
			break;
		}
	}
	return Date;
}

void PrintResult()
{
	stDate Date = ReadFullDate();

	short Days = ReadDaysToAdd();

	short TotalOfDays = 0;

	short DaysOrderInYear = NumberOfDaysFromTheBeginingOfTheYear(Date.Year,
Date.Month, Date.Day);

	TotalOfDays = DaysOrderInYear + Days;

	Date = DateAddDays(Date, TotalOfDays);

	cout << "\n\tDate after adding (" << Days << ") Days is : ";

	cout << Date.Day << "/" << Date.Month << "/" << Date.Year << endl;
}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
}
```

Problem #13: Date 1 Less Than Date 2

Write a program to read Date1, Date2 and Check if Date1 is Less Than Date2.

اكتب برنامجًا لقراءة التاريخ ١ والتاريخ ٢ والتحقق مما إذا كان التاريخ ١ أقل من التاريخ ٢ .

```
Enter a Date 1 :
Please Enter a day (1 to 31): 12
Please Enter a month (1 to 12): 2
PLease enter a year : 2022

Enter a Date 2 :
Please Enter a day (1 to 31): 12
Please Enter a month (1 to 12): 3
PLease enter a year : 2022

Yes, Date1 is less than Date2.
```

Solving:

```cpp
#include<iostream>
using namespace std;
struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month (1 to 12): ";
		cin >> m;
	} while (m < 1 || m > 12);
	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day (1 to 31): ";
		cin >> d;
	} while (d < 1 || d > 31);
	return d;
}
```

```
stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
	return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
	(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ?
	Date1.Day < Date2.Day : false)) : false);
}

void PrintResult()
{
	stDate Date1, Date2;

	cout << "\tEnter a Date 1 : " << endl;
	Date1 = ReadFullDate();

	cout << "\n\tEnter a Date 2 : " << endl;
	Date2 = ReadFullDate();

	if (IsDat1BeforDate2(Date1, Date2))
	{
		cout << "\n\tYes, Date1 is less than Date2." << endl;
	}
	else
		cout << "\n\tNo, Date1 is not less than Date2." << endl;
}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
}
```

Problem #14: Date 1 Equals To Date 2

Write a program to read Date1, Date2 and Check if Date1 Equals to Date2.

اكتب برنامجًا لقراءة التاريخ ١ والتاريخ ٢ والتحقق مما إذا كان التاريخ ١ يساوي التاريخ ٢.

```
Enter a Date 1 :
Please Enter a day (1 to 31): 1
Please Enter a month (1 to 12): 1
PLease enter a year : 2000

Enter a Date 2 :
Please Enter a day (1 to 31): 1
Please Enter a month (1 to 12): 1
PLease enter a year : 2000

Yes, Date1 is Equal To Date2.
```

Solving:

```
#include<iostream>
using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year: ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month (1 to 12): ";
		cin >> m;
	} while (m < 1 || m > 12);
	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day (1 to 31): ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}
```

```
stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

bool IsDat1EqualToDate2(stDate Date1, stDate Date2)
{
	return (Date1.Year == Date2.Year) ? ((Date1.Month == Date2.Month) ?
((Date1.Day == Date2.Day) ? true : false) : false) : false;
}

void PrintResult()
{
	stDate Date1, Date2;

	cout << "\tEnter a Date 1 : " << endl;
	Date1 = ReadFullDate();

	cout << "\n\tEnter a Date 2 : " << endl;
	Date2 = ReadFullDate();

	if (IsDat1EqualToDate2(Date1, Date2))
	{
		cout << "\n\tYes, Date1 is Equal To Date2." << endl;
	}
	else
		cout << "\n\tNo, Date1 is not Equal To Date2." << endl;
}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
}
```

Problem #15: Last Day, Last Month

Write a program to read a Date and Check:

1- if Day is last day in month.

2- if Month is last month in year.

اكتب برنامجًا لقراءة التاريخ والتحقق:

١- إذا كان اليوم هو آخر يوم في الشهر.

٢- إذا كان الشهر هو آخر شهر في السنة.

```
Please Enter a day (1 to 31): 31
Please Enter a month (1 to 12): 12
PLease enter a year : 2024

YES, Day Is Last Day In Month

YES, Month Is Last Month In Year
```

Solving:

```cpp
#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month (1 to 12): ";
		cin >> m;
	} while (m < 1 || m > 12);

	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day (1 to 31): ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}
```

```
stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

short NumberDaysInMonth(short year, short month)
{
	if (month < 1 || month > 12)
		return 0;

	short ArrMon[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

	return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsLastDayInMonth(stDate Date)
{
	return Date.Day == NumberDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{
	return (month == 12);
}

void PrintResult()
{
	stDate Date = ReadFullDate();

	if (IsLastDayInMonth(Date))
	{
		cout << "\n\tYES, Day Is Last Day In Month" << endl;
	}
	else
	{
		cout << "\n\tNO, Day Is Not Last Day In Month" << endl;
	}

	if (IsLastMonthInYear(Date.Month))
	{
		cout << "\n\tYES, Month Is Last Month In Year" << endl;
	}
	else
	{
		cout << "\n\tNO, Month Is Not Last Month In Year" << endl;
	}
}
```

```
int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
 }
```

==========

Problem #16: Increase Date By One Day

Write a program to read a date and make a function to increase date one day.

اكتب برنامجًا لقراءة التاريخ وقم بعمل دالة لزيادة التاريخ يومًا واحدًا.

```
	Please Enter a day (1 to 31): 31
	Please Enter a month (1 to 12): 12
	PLease enter a year : 2024

	Date after adding one day is: 1/1/2025
```

Solving:

```
#include<iostream>
using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{	short m;
	do
	{
		cout << "\tPlease Enter a month (1 to 12): ";
		cin >> m;
	} while (m < 1 || m > 12);
	return m;
}

short ReadDay()
{	short d;
	do
	{
		cout << "\tPlease Enter a day (1 to 31): ";
		cin >> d;
	} while (d < 1 || d > 31);
	return d;
}
```

```
stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

short NumberDaysInMonth(short year, short month)
{
	if (month < 1 || month > 12)
		return 0;

	short ArrMon[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

	return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsLastDayInMonth(stDate Date)
{
	return Date.Day == NumberDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{
	return (month == 12);
}

stDate IncreaseDateByOneDay(stDate Date)
{
	if (IsLastDayInMonth(Date))
	{
		if(IsLastMonthInYear(Date.Month))
		{
			Date.Day = 1;
			Date.Month = 1;
			Date.Year++;
		}
		else
		{
			Date.Day = 1;
			Date.Month++;
		}
	}
	else
	{
		Date.Day++;
	}
	return Date;
}
```

```
void PrintResult()
{
	stDate Date = ReadFullDate();

	Date = IncreaseDateByOneDay(Date);

	cout << "\n\tDate after adding one day is: " << Date.Day << "/" <<
Date.Month << "/" << Date.Year << endl;
}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
 }
```

==========

Problem #17: Diff In Days

Write a program to read a Date1, Date2 and make a function to calculate the difference in days.

Note: Date1 should be less than Date2.

اكتب برنامجًا لقراءة التاريخ ١ والتاريخ ٢ وعمل دالة لحساب الفرق في الأيام.

ملاحظة: يجب أن يكون Date1 أقل من Date2.

```
Enter a Date 1:
-------------------------
Please Enter a day  : 1
Please Enter a month: 1
PLease enter a year : 2024

Enter a Date 2:
Please Enter a day  : 25
Please Enter a month: 5
PLease enter a year : 2024
-------------------------

Diffrence is : 145 Day(s).

Difference(Includeing End Day) is : 146 Day(s).
```

Solving:

```
#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};
```

```
short ReadYear()
{     short y;
      cout << "\tPLease enter a year : ";
      cin >> y;
      return y;
}

short ReadMonth()
{     short m;
      do
      {
            cout << "\tPlease Enter a month (1 to 12): ";
            cin >> m;
      } while (m < 1 || m > 12);
      return m;
}

short ReadDay()
{     short d;
      do
      {
            cout << "\tPlease Enter a day (1 to 31): ";
            cin >> d;
      } while (d < 1 || d > 31);
      return d;
}

stDate ReadFullDate()
{
      stDate Date;
      Date.Day = ReadDay();
      Date.Month = ReadMonth();
      Date.Year = ReadYear();
      return Date;
}

bool IsLeapYear(short year)
{
      return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
      return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day <
Date2.Day : false)) : false);
}

short NumberOfDaysInMonth(short year, short month)
{
      if (month < 1 || month > 12)
            return 0;
      short ArrMon[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

      return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}
```

```
bool IsLastDayInMonth(stDate Date)
{
	return Date.Day == NumberOfDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{
	return (month == 12);
}

stDate IncreaseDateByOneDay(stDate Date)
{
	if (IsLastDayInMonth(Date))
	{
		if(IsLastMonthInYear(Date.Month))
		{
			Date.Day = 1;
			Date.Month = 1;
			Date.Year++;
		}
		else
		{
			Date.Day = 1;
			Date.Month++;
		}
	}
	else
	{
		Date.Day++;
	}
	return Date;
}

int GetDifferenceInDays(stDate Date1, stDate Date2, bool includeEndDay = false)
{
	short days = 0;
	while (IsDat1BeforDate2(Date1, Date2))
	{
		days++;
		Date1 = IncreaseDateByOneDay(Date1);
	}
	return includeEndDay ? ++days : days;
}

void PrintResult()
{
	cout << "\tEnter Date 1 : " << endl;
	stDate Date1 = ReadFullDate();

	cout << "\n\tEnter Date 2 : " << endl;
	stDate Date2 = ReadFullDate();
	cout << "\t_____________________________" << endl;
	cout << "\n\tDiffrence is : " << GetDifferenceInDays(Date1, Date2) << " Day(s)." << endl;
	cout << "\n\tDiffrence(Includeing End Day) is: " <<
		GetDifferenceInDays(Date1, Date2, true) << " Day(s)." << endl;

}
```

```
int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
 }
```

==========

Problem #18: Your Age In Days

Write a program calculate your age in days.

أكتب برنامج يحسب عمرك بالأيام.

```
   Enter Your Date Of Birth :
   -------------------------
   Please Enter a day  : 3
   Please Enter a month: 5
   PLease enter a year : 2001
   -------------------------

   Your Age is : 8423 Day(s).
```

Solving:

```
#pragma warning(disable : 4996)
#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);

	return m;
}
```

```
short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true :
false;
}

bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
	return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day <
Date2.Day : false)) : false);
}

short NumberOfDaysInMonth(short year, short month)
{
	if (month < 1 || month > 12)
		return 0;

	short ArrMon[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

	return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsLastDayInMonth(stDate Date)
{
	return Date.Day == NumberOfDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{
	return (month == 12);
}
```

```
stDate IncreaseDateByOneDay(stDate Date)
{
	if (IsLastDayInMonth(Date))
	{
		if(IsLastMonthInYear(Date.Month))
		{
			Date.Day = 1;
			Date.Month = 1;
			Date.Year++;
		}
		else
		{
			Date.Day = 1;
			Date.Month++;
		}
	}
	else
	{
		Date.Day++;
	}
	return Date;
}

int GetDifferenceInDays(stDate Date1, stDate Date2, bool includeEndDay = false)
{
	short days = 0;

	while (IsDat1BeforDate2(Date1, Date2))
	{
		days++;
		Date1 = IncreaseDateByOneDay(Date1);
	}
	return includeEndDay ? ++days : days;
}

stDate GetSystemDate()
{
	stDate Date;
	time_t t = time(0);
	tm* now = localtime(&t);
	Date.Year = now->tm_year + 1900;
	Date.Month = now->tm_mon + 1;
	Date.Day = now->tm_mday;

	return Date;
}

void PrintResult()
{
	cout << "\tEnter Your Date Of Birth : " << endl;
	cout << "\t--------------------------\n";
	stDate Date1 = ReadFullDate();
	stDate Date2 = GetSystemDate();
	cout << "\t--------------------------\n";
	cout << "\n\tYour Age is : " << GetDifferenceInDays(Date1, Date2, true)
		<< " Day(s)." << endl;
}
```

```
int main()
{
	system("color f0");
	PrintResult();
	system("pause>0");
	return 0;
 }
```

==========

Problem #19: Diff In Days (Negative Days)

Write a program to read a Date1, Date2 and make a function to calculate the difference in days.
Note: if Date2 is less than Date1 print the result in minus.

اكتب برنامجًا لقراءة التاريخ ١ والتاريخ ٢ وعمل دالة لحساب الفرق في الأيام.
ملاحظة: إذا كان Date2 أقل من Date1، فاطبع النتيجة بالناقص.

```
Enter a Date 1:
-------------------------
Please Enter a day  : 25
Please Enter a month: 5
PLease enter a year : 2024

Enter a Date 2:
Please Enter a day  : 1
Please Enter a month: 1
PLease enter a year : 2024
-------------------------

Diffrence is : -145 Day(s).

Difference(Includeing End Day) is : -146 Day(s).
```

Solving:

```
#include<iostream>
using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);
	return m;
}
```

```
short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true :
false;
}

bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
	return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day <
Date2.Day : false)) : false);
}

short NumberOfDaysInMonth(short year, short month)
{
	if (month < 1 || month > 12)
		return 0;

	short ArrMon[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

	return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsLastDayInMonth(stDate Date)
{
	return Date.Day == NumberOfDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{
	return (month == 12);
}
```

```
stDate IncreaseDateByOneDay(stDate Date)
{
	if (IsLastDayInMonth(Date))
	{
		if(IsLastMonthInYear(Date.Month))
		{
			Date.Day = 1;
			Date.Month = 1;
			Date.Year++;
		}
		else
		{
			Date.Day = 1;
			Date.Month++;
		}
	}
	else
	{
		Date.Day++;
	}
	return Date;
}

void SwapDates(stDate &Date1, stDate &Date2)
{
	stDate TampDate;
	TampDate.Day = Date1.Day;
	TampDate.Month = Date1.Month;
	TampDate.Year = Date1.Year;

	Date1.Day = Date2.Day;
	Date1.Month = Date2.Month;
	Date1.Year = Date2.Year;

	Date2.Day = TampDate.Day;
	Date2.Month = TampDate.Month;
	Date2.Year = TampDate.Year;
}

int GetDifferenceInDays(stDate Date1, stDate Date2, bool includeEndDay = false)
{
	short days = 0;
	short SwapFlagValue = 1;
	if (!IsDat1BeforDate2(Date1, Date2))
	{
		SwapDates(Date1, Date2);
		SwapFlagValue = -1;
	}

	while (IsDat1BeforDate2(Date1, Date2))
	{
		days++;
		Date1 = IncreaseDateByOneDay(Date1);
	}

	return includeEndDay ? ++days * SwapFlagValue: days * SwapFlagValue;
}
```

```
void PrintResult()
{
	cout << "\tEnter a Date 1: " << endl;
	cout << "\t-------------------------\n";
	stDate Date1 = ReadFullDate();
	cout << "\n\tEnter a Date 2: " << endl;
	stDate Date2 = ReadFullDate();

	cout << "\t-------------------------\n";

	cout << "\n\tDiffrence is : " << GetDifferenceInDays(Date1, Date2) << "
Day(s)." << endl;
	cout << "\n\tDifference(Includeing End Day) is : " <<
GetDifferenceInDays(Date1, Date2, true) << " Day(s)." << endl;

}

int main()
{
	system("color f0");
	PrintResult();
	system("pause>0");
	return 0;
 }
```

==========

Problem #20 - #32 - Increase Date Problems (20 - 32)

Write a program to read a Date and make a functions to increase date as follows:

قم بكتابة برنامج لقراءة التاريخ وقم بعمل دالة لزيادة التاريخ كما يلي:

- **IncreaseDateByOneDay.**
- **IncreaseDateByXDays.**
- **IncreaseDateByOneWeek.**
- **IncreaseDateByXWeeks.**
- **IncreaseDateByOneMonth.**
- **IncreaseDateByXMonths.**
- **IncreaseDateByOneYear.**
- **IncreaseDateByXYears.**
- **IncreaseDateByXYearsFaster.**
- **IncreaseDateByOneDecade.**
- **IncreaseDateByXDecades.**
- **IncreaseDateByXDecadesFaster.**
- **IncreaseDateByOneCentury.**
- **IncreaseDateByOneMillennium.**

```
Please Enter a day  : 31
Please Enter a month: 12
PLease enter a year : 2024

Date After:
-----------------------------------
01-Adding One Day is  : 1/1/2025
02-Adding 10 Days is  : 11/1/2025
03-Adding One Week is : 18/1/2025
04-Adding 10 Weeks is : 29/3/2025
05-Adding One Month is: 29/4/2025
06-Adding 5 Months is : 29/9/2025
07-Adding One Year is : 29/9/2026
08-Adding 10 Year is  : 29/9/2036
09-Adding 10 Year (Faster) is   : 29/9/2046
10-Adding One Decade is: 29/9/2056
11-Adding 10 Decades is: 29/9/2156
12-Adding 10 Decades (Faster) is: 29/9/2256
12-Adding One Century is   : 29/9/2356
12-Adding One Millennium is: 29/9/3356
```

Solving:

```
#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};
```

```
short EnterDays()
{
	short Days;
	cout << "\tEnter how many days you want to add: ";
	cin >> Days;
	return Days;
}

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);

	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true :
false;
}
```

```
short NumberDaysInMonth(short year, short month)
{
     if (month < 1 || month > 12)
          return 0;
     short ArrMon[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
     return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsLastDayInMonth(stDate Date)
{
     return Date.Day == NumberDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{    return (month == 12);
}

stDate IncreaseDateByOneDay(stDate Date)
{
     if (IsLastDayInMonth(Date))
     {
          if (IsLastMonthInYear(Date.Month))
          {
               Date.Day = 1;
               Date.Month = 1;
               Date.Year++;
          }
          else
          {
               Date.Day = 1;
               Date.Month++;
          }
     }
     else
     {
          Date.Day++;
     }
     return Date;
}

stDate IncreaseDateByXDays(stDate Date, short Days)
{
     for(short i = 1; i<= Days; i++)
     {
          Date = IncreaseDateByOneDay(Date);
     }
     return Date;
}

stDate IncreaseDateByOneWeek(stDate Date)
{    short Week = 7;
     for(short i = 1; i <= Week; i++)
     {
          Date = IncreaseDateByOneDay(Date);
     }
     return Date;
}
```

```
stDate IncreaseDateByXWeek(stDate Date, short Weeks)
{
      for (short i = 1; i <= Weeks; i++)
      {
            Date = IncreaseDateByOneWeek(Date);
      }
      return Date;
}

stDate IncreaseDateByOneMonth(stDate Date)
{
            if (Date.Month == 12)
            {
                  Date.Month = 1;
                  Date.Year++;
            }
            else
            {
                  Date.Month++;
            }
            /* Last check day in date should not exceed max days in the current month
               example if date is 31/1/2024 increasing one month
               should not be 31/2/2024, it should be 28/2/2024 */
            short NumberOfDaysInCurrentMonth = NumberDaysInMonth(Date.Year,
Date.Month);
            if (Date.Day > NumberOfDaysInCurrentMonth)
            {
                  Date.Day = NumberOfDaysInCurrentMonth;
            }
      return Date;
}

stDate IncreaseDateByXMonth(stDate Date, short M)
{
      for (short i = 1; i <= M; i++)
      {
            Date = IncreaseDateByOneMonth(Date);
      }
      return Date;
}

stDate IncreaseDateByOneYear(stDate Date)
{
      Date.Year++;
      return Date;
}

stDate IncreaseDateByXYear(stDate Date, short Y)
{
      for (short i = 1; i <= Y; i++)
      {
            Date = IncreaseDateByOneYear(Date);
      }
      return Date;
}
```

```
stDate IncreaseDateByXYearFaster(stDate Date, short Y) // Optimise
{
	Date.Year += Y;

	return Date;
}

stDate IncreaseDateByOneDecade(stDate Date)
{
	//Period of 10 years
	Date.Year += 10;

	return Date;
}

stDate IncreaseDateByXDecades(stDate Date, short Decade)
{
	for (short i = 1; i <= Decade * 10; i++)
	{
		Date = IncreaseDateByOneYear(Date);
	}
	return Date;
}

stDate IncreaseDateByXDecadesFaster(stDate Date, short Decade) // Optimise
{
	Date.Year += Decade * 10;

	return Date;
}

stDate IncreaseDateByOneCentury(stDate Date)
{
	//Period of 100 years
	Date.Year += 100;

	return Date;
}

stDate IncreaseDateByOneMillennium(stDate Date)
{
	//Period of 1000 years
	Date.Year += 1000;
	return Date;
}

void PrintResult()
{
	stDate Date = ReadFullDate();
	cout << "\n\tDate After: \n";
	cout << "\t-----------------------------------";
	Date = IncreaseDateByOneDay(Date);
	cout << "\n\t01-Adding One Day is  : " << Date.Day << "/" << Date.Month <<
"/" << Date.Year << endl;
	Date = IncreaseDateByXDays(Date, 10);
	cout << "\t02-Adding 10 Days is  : " << Date.Day << "/" << Date.Month <<
"/" << Date.Year << endl;
```

```
	Date = IncreaseDateByOneWeek(Date);
	cout << "\t03-Adding One Week is : " << Date.Day << "/" << Date.Month <<
"/" << Date.Year << endl;

	Date = IncreaseDateByXWeek(Date, 10);
	cout << "\t04-Adding 10 Weeks is : " << Date.Day << "/" << Date.Month <<
"/" << Date.Year << endl;

	Date = IncreaseDateByOneMonth(Date);
	cout << "\t05-Adding One Month is: " << Date.Day << "/" << Date.Month <<
"/" << Date.Year << endl;

	Date = IncreaseDateByXMonth(Date, 5);
	cout << "\t06-Adding 5 Months is : " << Date.Day << "/" << Date.Month <<
"/" << Date.Year << endl;

	Date = IncreaseDateByOneYear(Date);
	cout << "\t07-Adding One Year is : " << Date.Day << "/" << Date.Month <<
"/" << Date.Year << endl;

	Date = IncreaseDateByXYear(Date, 10);
	cout << "\t08-Adding 10 Year is  : " << Date.Day << "/" << Date.Month <<
"/" << Date.Year << endl;

	Date = IncreaseDateByXYearFaster(Date, 10);
	cout << "\t09-Adding 10 Year (Faster) is   : " << Date.Day << "/" <<
Date.Month << "/" << Date.Year << endl;

	Date = IncreaseDateByOneDecade(Date);
	cout << "\t10-Adding One Decade is: " << Date.Day << "/" << Date.Month <<
"/" << Date.Year << endl;

	Date = IncreaseDateByXDecades(Date, 10);
	cout << "\t11-Adding 10 Decades is: " << Date.Day << "/" << Date.Month <<
"/" << Date.Year << endl;

	Date = IncreaseDateByXDecadesFaster(Date, 10);
	cout << "\t12-Adding 10 Decades (Faster) is: " << Date.Day << "/" <<
Date.Month << "/" << Date.Year << endl;

	Date = IncreaseDateByOneCentury(Date);
	cout << "\t13-Adding One Century is   : " << Date.Day << "/" << Date.Month
<< "/" << Date.Year << endl;

	Date = IncreaseDateByOneMillennium(Date);
	cout << "\t14-Adding One Millennium is: " << Date.Day << "/" << Date.Month
<< "/" << Date.Year << endl;
}

int main()
{
	system("color f0");

	PrintResult();
	system("pause>0");
	return 0;
}
```

Problem #33 - #46 - Decrease Date Problems (33 - 46)

Write a program to read a Date and make a functions to Decrease date as follows:

قم بكتابة برنامج لقراءة التاريخ وعمل دالة لتقليل التاريخ كما يلي:

- DecreaseDateByOneDay
- DecreaseDateByXDays.
- DecreaseDateByOneWeek.
- DecreaseDateByXWeeks.
- DecreaseDateByOneMonth.
- DecreaseDateByXMonths.
- DecreaseDateByOneYear.
- DecreaseDateByXYears.
- DecreaseDateByXYearsFaster.
- DecreaseDateByOneDecade.
- DecreaseDateByXDecades.
- DecreaseDateByXDecadesFaster.
- DecreaseDateByOneCentury.
- DecreaseDateByOneMillennium.

```
Please Enter a day  : 31
Please Enter a month: 12
PLease enter a year : 2024

Date After:
-----------------------------------
01-Subtracting One Day is  : 30/12/2024
02-Subtracting 10 Days is  : 20/12/2024
03-Subtracting One Week is : 13/12/2024
04-Subtracting 10 Weeks is : 4/10/2024
05-Subtracting One Month is: 4/9/2024
06-Subtracting 5 Months is : 4/4/2024
07-Subtracting One Year is : 4/4/2023
08-Subtracting 10 Year is  : 4/4/2013
09-Subtracting 10 Year (Faster) is   : 4/4/2003
10-Subtracting One Decade is: 4/4/1993
11-Subtracting 10 Decades is: 4/4/1893
12-Subtracting 10 Decades (Faster) is: 4/4/1793
13-Subtracting One Century is   : 4/4/1693
14-Subtracting One Millennium is: 4/4/693
```

Solving:

```
#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short EnterDays()
{
	short Days;
	cout << "\tEnter how many days you want to add: ";
	cin >> Days;
	return Days;
}

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);
	return m;
}
```

```
short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

short NumberDaysInMonth(short year, short month)
{
	if (month < 1 || month > 12)
		return 0;

	short ArrMon[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
	return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

stDate DecreaseDateByOneDay(stDate Date)
{
	if (Date.Day == 1)
	{
		if (Date.Month == 1)
		{
			Date.Day = 31;
			Date.Month = 12;
			Date.Year--;
		}
		else
		{
			Date.Month--;
			Date.Day = NumberDaysInMonth(Date.Year, Date.Month);
		}
	}
	else
	{
		Date.Day--;
	}
	return Date;
}
```

```
stDate DecreaseDateByXDays(stDate Date, short Days)
{
	for (short i = 1; i <= Days; i++)
	{
		Date = DecreaseDateByOneDay(Date);
	}
	return Date;
}

stDate DecreaseDateByOneWeek(stDate Date)
{
	for(short i = 1; i <= 7; i++)
	{
		Date = DecreaseDateByOneDay(Date);
	}
	return Date;
}

stDate DecreaseDateByXWeek(stDate Date, short Weeks)
{
	for (short i = 1; i <= Weeks; i++)
	{
		Date = DecreaseDateByOneWeek(Date);
	}
	return Date;
}

stDate DecreaseDateByOneMonth(stDate Date)
{
	if (Date.Month == 1)
	{
		Date.Month = 12;
		Date.Year--;
	}
	else
	{
		Date.Month--;
	}
	/* Last check day in date should not exceed max days in the current month
		example if date is 31/3/2024 decreasing one month
		should not be 31/2/2024, it should be 28/2/2024 */
	short NumberOfDaysInCurrentMonth = NumberDaysInMonth(Date.Year, Date.Month);
	if (Date.Day > NumberOfDaysInCurrentMonth)
	{
		Date.Day = NumberOfDaysInCurrentMonth;
	}
	return Date;
}

stDate DecreaseDateByXMonth(stDate Date, short M)
{
	for (short i = 1; i <= M; i++)
	{
		Date = DecreaseDateByOneMonth(Date);
	}
	return Date;
}
```

```
stDate DecreaseDateByOneYear(stDate Date)
{
	Date.Year--;
	return Date;
}

stDate DecreaseDateByXYear(stDate Date, short Y)
{
	for (short i = 1; i <= Y; i++)
	{
		Date = DecreaseDateByOneYear(Date);
	}
	return Date;
}

stDate DecreaseDateByXYearFaster(stDate Date, short Y) // Optimise
{
	Date.Year -= Y;
	return Date;
}

stDate DecreaseDateByOneDecade(stDate Date)
{
	//Period of 10 years
	Date.Year -= 10;
	return Date;
}

stDate DecreaseDateByXDecades(stDate Date, short Decade)
{
	for (short i = 1; i <= Decade * 10; i++)
	{
		Date = DecreaseDateByOneYear(Date);
	}
	return Date;
}

stDate DecreaseDateByXDecadesFaster(stDate Date, short Decade) // Optimise
{
	Date.Year -= Decade * 10;
	return Date;
}

stDate DecreaseDateByOneCentury(stDate Date)
{
	//Period of 100 years
	Date.Year -= 100;
	return Date;
}

stDate DecreaseDateByOneMillennium(stDate Date)
{
	//Period of 1000 years
	Date.Year -= 1000;

	return Date;
}
```

```
void PrintResult()
{
     stDate Date = ReadFullDate();
     cout << "\n\tDate After: \n";
     cout << "\t-----------------------------------";
     Date = DecreaseDateByOneDay(Date);
     cout << "\n\t01-Subtracting One Day is  : " << Date.Day << "/" <<
Date.Month << "/" << Date.Year << endl;

     Date = DecreaseDateByXDays(Date, 10);
     cout << "\t02-Subtracting 10 Days is  : " << Date.Day << "/" << Date.Month
<< "/" << Date.Year << endl;

     Date = DecreaseDateByOneWeek(Date);
     cout << "\t03-Subtracting One Week is : " << Date.Day << "/" << Date.Month
<< "/" << Date.Year << endl;

     Date = DecreaseDateByXWeek(Date, 10);
     cout << "\t04-Subtracting 10 Weeks is : " << Date.Day << "/" << Date.Month
<< "/" << Date.Year << endl;

     Date = DecreaseDateByOneMonth(Date);
     cout << "\t05-Subtracting One Month is: " << Date.Day << "/" << Date.Month
<< "/" << Date.Year << endl;

     Date = DecreaseDateByXMonth(Date, 5);
     cout << "\t06-Subtracting 5 Months is : " << Date.Day << "/" << Date.Month
<< "/" << Date.Year << endl;

     Date = DecreaseDateByOneYear(Date);
     cout << "\t07-Subtracting One Year is : " << Date.Day << "/" << Date.Month
<< "/" << Date.Year << endl;

     Date = DecreaseDateByXYear(Date, 10);
     cout << "\t08-Subtracting 10 Year is  : " << Date.Day << "/" << Date.Month
<< "/" << Date.Year << endl;

     Date = DecreaseDateByXYearFaster(Date, 10);
     cout << "\t09-Subtracting 10 Year (Faster) is   : " << Date.Day << "/" <<
Date.Month << "/" << Date.Year << endl;

     Date = DecreaseDateByOneDecade(Date);
     cout << "\t10-Subtracting One Decade is: " << Date.Day << "/" <<
Date.Month << "/" << Date.Year << endl;

     Date = DecreaseDateByXDecades(Date, 10);
     cout << "\t11-Subtracting 10 Decades is: " << Date.Day << "/" <<
Date.Month << "/" << Date.Year << endl;

     Date = DecreaseDateByXDecadesFaster(Date, 10);
     cout << "\t12-Subtracting 10 Decades (Faster) is: " << Date.Day << "/" <<
Date.Month << "/" << Date.Year << endl;

     Date = DecreaseDateByOneCentury(Date);
     cout << "\t13-Subtracting One Century is   : " << Date.Day << "/" <<
Date.Month << "/" << Date.Year << endl;
```

```
	Date = DecreaseDateByOneMillennium(Date);
	cout << "\t14-Subtracting One Millennium is: " << Date.Day << "/" <<
Date.Month << "/" << Date.Year << endl;
}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
}
```

==========

Problem #47 - #53 : More Date Problems (47 - 53)

Write a program to read a Date and make functions as follows:

اكتب برنامجًا لقراءة التاريخ وإجراء الوظائف كما يلي:

- **Overload the DayOfWeekOrder to take date structure**
- **IsEndOfWeek**
- **IsWeekEnd**
- **IsBusinessDay**
- **DaysUntilTheEndOfWeek**
- **DaysUntilTheEndOfMonth**
- **DaysUntilTheEndOfYear**

```
Today is : Tue , 28/5/2024

Is it End of Week ?
No, Not end of week.

Is it Weekend? ?
No, it is Not a week end.

Is it Business Day ?
Yes, it is a business day.

Days Until The End Of Week : 4 Day(s)

Days Until The End Of Month: 4 Day(s)

Days Until The End Of Year : 218 Day(s)
```

Solving:

```
#pragma warning (disable : 4996)

#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short EnterDays()
{
	short Days;
	cout << "\tEnter how many days you want to add: ";
	cin >> Days;
	return Days;
}

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}
```

```
short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);

	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

short NumberDaysInMonth(short year, short month)
{
	if (month < 1 || month > 12)
		return 0;

	short ArrMon[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

	return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
	return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day <
Date2.Day : false)) : false);
}
```

```
bool IsLastDayInMonth(stDate Date)
{
     return Date.Day == NumberDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{
     return (month == 12);
}

stDate IncreaseDateByOneDay(stDate Date)
{
     if (IsLastDayInMonth(Date))
     {
          if (IsLastMonthInYear(Date.Month))
          {
               Date.Day = 1;
               Date.Month = 1;
               Date.Year++;
          }
          else
          {
               Date.Day = 1;
               Date.Month++;
          }
     }
     else
     {
          Date.Day++;
     }
     return Date;
}

int GetDifferenceInDays(stDate Date1, stDate Date2, bool includeEndDay = false)
{
     short days = 0;

     while (IsDat1BeforDate2(Date1, Date2))
     {
          days++;
          Date1 = IncreaseDateByOneDay(Date1);
     }
     return includeEndDay ? ++days : days;
}

short DayOfWeekOrder(short Year, short Month, short Day)
{
     short a, y, m;
     a = ((14 - Month) / 12);
     y = Year - a;
     m = Month + (12 * a) - 2;
     // Greforian:
     // The value of d is 0 for a Sunday, 1 for a Monday, 2 for a Tuesday etc..
     return (Day + y + (y / 4) - (y / 100) + (y / 400) + ((31 * m) / 12)) % 7;

}
```

```
short DayOfWeekOrder(stDate Date)
{
	return DayOfWeekOrder(Date.Year, Date.Month, Date.Day);
}

string DayShortName(short DayOfWeekOrder)
{
	string arrDayName[7] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};
	return arrDayName[DayOfWeekOrder];
}

stDate GetSystemDate()
{
	stDate Date;
	time_t t = time(0);
	tm* now = localtime(&t);

	Date.Year = now->tm_year + 1900;
	Date.Month = now->tm_mon + 1;
	Date.Day = now->tm_mday;

	return Date;
}

bool IsEndOfWeek(stDate Date)
{
	return DayOfWeekOrder(Date) == 6;
}

bool IsWeekEnd(stDate Date)
{
	return DayOfWeekOrder(Date) == 5;
}

bool IsBusinessDay(stDate Date)
{
	return !IsWeekEnd(Date);
}

short DaysUntilTheEndOfWeek(stDate Date)
{
	return 6 - DayOfWeekOrder(Date);
}

short DaysUntilTheEndOfMonth(stDate Date)
{
	stDate EndOfMonthDate;

	EndOfMonthDate.Day = NumberDaysInMonth(Date.Year, Date.Month);

	EndOfMonthDate.Month = Date.Month;

	EndOfMonthDate.Year = Date.Year;

	return GetDifferenceInDays(Date, EndOfMonthDate, true);
}
```

```
short DaysUntilTheEndOfYear(stDate Date)
{
	stDate EndOfYearDate;
	EndOfYearDate.Day = 31;
	EndOfYearDate.Month = 12;
	EndOfYearDate.Year = Date.Year;
	return GetDifferenceInDays(Date, EndOfYearDate, true);
}

void PrintIsEndOfWeek(stDate Date)
{
	cout << "\n\tIs it End of Week ?\n";
	if (IsEndOfWeek(Date))
		cout << "\tYes, it is end of week.\n";
	else
		cout << "\tNo, Not end of week.\n";
}

void PrintIsWeekEnd(stDate Date)
{
	cout << "\n\tIs it Weekend? ?\n";
	if (IsWeekEnd(Date))
		cout << "\tYes, it is a week end.\n";
	else
		cout << "\tNo, it is Not a week end.\n";
}

void PrintIsBusinessDay(stDate Date)
{
	cout << "\n\tIs it Business Day ?\n";
	if (IsBusinessDay(Date))
		cout << "\tYes, it is a business day.\n";
	else
		cout << "\tNo, it is NOT a business day.\n";
}

void PrintDaysUntilTheEndOfWeek(stDate Date)
{
	cout << "\n\tDays Until The End Of Week : " << DaysUntilTheEndOfWeek(Date)
<< " Day(s)" << endl;
}

void PrintDaysUntilTheEndOfMonth(stDate Date)
{
	cout << "\n\tDays Until The End Of Month: " <<
DaysUntilTheEndOfMonth(Date) << " Day(s)" << endl;
}

void PrintDaysUntilTheEndOfYear(stDate Date)
{
	cout << "\n\tDays Until The End Of Year : " << DaysUntilTheEndOfYear(Date)
<< " Day(s)" << endl;
}
```

```
void PrintResult()
{
	//stDate Date1 = ReadFullDate();
	stDate Date1 = GetSystemDate();

	cout << "\n\tToday is : ";
	cout << DayShortName(DayOfWeekOrder(Date1)) << " , ";
	cout << Date1.Day << "/" << Date1.Month << "/" << Date1.Year << endl;

	PrintIsEndOfWeek(Date1);

	PrintIsWeekEnd(Date1);

	PrintIsBusinessDay(Date1);

	PrintDaysUntilTheEndOfWeek(Date1);

	PrintDaysUntilTheEndOfMonth(Date1);

	PrintDaysUntilTheEndOfYear(Date1);
}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
}
```

==========

Problem #54 - Calculate Vacation Days

Write a program to read Vacation Period (Date From and Date To) and make a function to calculate the actual vacation days.

Note: Weekends are excluded.

اكتب برنامجًا لقراءة فترة الإجازة (تاريخ من وتاريخ إلى) وقم بعمل دالة لحساب أيام الإجازة الفعلية.

ملاحظة: يتم استبعاد عطلات نهاية الأسبوع.

```
Vacation Starts:
Please Enter a day  : 1
Please Enter a month: 6
PLease enter a year : 2024

Vacation Ends:
Please Enter a day  : 5
Please Enter a month: 6
PLease enter a year : 2024

Vacation From : Sat , 1/6/2024
Vacation To   : Wed , 5/6/2024

Actucal Vacation Days is: 3
```

Solving:

```
#pragma warning (disable : 4996)

#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);

	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}
```

```
bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

short NumberDaysInMonth(short year, short month)
{
	if (month < 1 || month > 12)
		return 0;

	short ArrMon[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

	return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
	return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day <
Date2.Day : false)) : false);
}

bool IsLastDayInMonth(stDate Date)
{
	return Date.Day == NumberDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{
	return (month == 12);
}

stDate IncreaseDateByOneDay(stDate Date)
{
	if (IsLastDayInMonth(Date))
	{
		if (IsLastMonthInYear(Date.Month))
		{
			Date.Day = 1;
			Date.Month = 1;
			Date.Year++;
		}
		else
		{
			Date.Day = 1;
			Date.Month++;
		}
	}
	else
	{
		Date.Day++;
	}
	return Date;
}
```

```
int GetDifferenceInDays(stDate Date1, stDate Date2, bool includeEndDay = false)
{
	short days = 0;

	while (IsDat1BeforDate2(Date1, Date2))
	{
		days++;
		Date1 = IncreaseDateByOneDay(Date1);
	}
	return includeEndDay ? ++days : days;
}

short DayOfWeekOrder(short Year, short Month, short Day)
{
	short a, y, m;
	a = ((14 - Month) / 12);
	y = Year - a;
	m = Month + (12 * a) - 2;
	// Greforian:
	// The value of d is 0 for a Sunday, 1 for a Monday, 2 for a Tuesday etc..
	return (Day + y + (y / 4) - (y / 100) + (y / 400) + ((31 * m) / 12)) % 7;

}

short DayOfWeekOrder(stDate Date)
{
	return DayOfWeekOrder(Date.Year, Date.Month, Date.Day);
}

string DayShortName(short DayOfWeekOrder)
{
	string arrDayName[7] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};
	return arrDayName[DayOfWeekOrder];
}

stDate GetSystemDate()
{
	stDate Date;
	time_t t = time(0);
	tm* now = localtime(&t);

	Date.Year = now->tm_year + 1900;
	Date.Month = now->tm_mon + 1;
	Date.Day = now->tm_mday;

	return Date;
}

bool IsEndOfWeek(stDate Date)
{
	return DayOfWeekOrder(Date) == 6;
}

bool IsWeekEnd(stDate Date)
{
	return (DayOfWeekOrder(Date) == 5 || DayOfWeekOrder(Date) == 6);
}
```

```
bool IsBusinessDay(stDate Date)
{
	return !IsWeekEnd(Date);
}

short CalculatelVacationDays(stDate DateFrom, stDate DateTo)
{
	short DaysCount = 0;

	while (IsDat1BeforDate2(DateFrom, DateTo))
	{
		if (IsBusinessDay(DateFrom))
			DaysCount++;

		DateFrom = IncreaseDateByOneDay(DateFrom);
	}
	return DaysCount;
}

void PrintResult()
{
	cout << "\tVacation Starts:" << endl;
	stDate DateFrom = ReadFullDate();

	cout << "\n\tVacation Ends:" << endl;
	stDate DateTo = ReadFullDate();


	cout << "\n\tVacation From : ";
	cout << DayShortName(DayOfWeekOrder(DateFrom)) << " , ";
	cout << DateFrom.Day << "/" << DateFrom.Month << "/" << DateFrom.Year <<
endl;

	cout << "\tVacation To   : ";
	cout << DayShortName(DayOfWeekOrder(DateTo)) << " , ";
	cout << DateTo.Day << "/" << DateTo.Month << "/" << DateTo.Year << endl;

	cout << "\n\tActucal Vacation Days is: " <<
CalculatelVacationDays(DateFrom, DateTo) << endl;

}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
}
```

Problem #55 - Calculate Vacation Return Date

Write a program to read Vacation Start (Date From and Vacation Days), then make a function to calculate the vacation return date.
Note: Weekends are excluded.

اكتب برنامجًا لقراءة بداية الإجازة (التاريخ من وأيام الإجازة)، ثم قم بعمل دالة لحساب تاريخ عودة الإجازة.
ملاحظة: يتم استبعاد عطلات نهاية الأسبوع.

```
Vacation Starts:
Please Enter a day  : 1
Please Enter a month: 6
PLease enter a year : 2024

Please enter Vacation days: 10

Return Date: Sun , 16/6/2024
```

Solving:

```
#pragma warning (disable : 4996)
#include<iostream>
using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);
	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);
	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}
```

```
bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

short NumberDaysInMonth(short year, short month)
{
	if (month < 1 || month > 12)
		return 0;

	short ArrMon[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

	return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
	return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day <
Date2.Day : false)) : false);
}

bool IsLastDayInMonth(stDate Date)
{
	return Date.Day == NumberDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{
	return (month == 12);
}

stDate IncreaseDateByOneDay(stDate Date)
{
	if (IsLastDayInMonth(Date))
	{
		if (IsLastMonthInYear(Date.Month))
		{
			Date.Day = 1;
			Date.Month = 1;
			Date.Year++;
		}
		else
		{
			Date.Day = 1;
			Date.Month++;
		}
	}
	else
	{
		Date.Day++;
	}
	return Date;
}
```

```
int GetDifferenceInDays(stDate Date1, stDate Date2, bool includeEndDay = false)
{
	short days = 0;

	while (IsDat1BeforDate2(Date1, Date2))
	{
		days++;
		Date1 = IncreaseDateByOneDay(Date1);
	}
	return includeEndDay ? ++days : days;
}

short DayOfWeekOrder(short Year, short Month, short Day)
{
	short a, y, m;
	a = ((14 - Month) / 12);
	y = Year - a;
	m = Month + (12 * a) - 2;
	// Greforian:
	// The value of d is 0 for a Sunday, 1 for a Monday, 2 for a Tuesday etc..
	return (Day + y + (y / 4) - (y / 100) + (y / 400) + ((31 * m) / 12)) % 7;

}

short DayOfWeekOrder(stDate Date)
{
	return DayOfWeekOrder(Date.Year, Date.Month, Date.Day);
}

string DayShortName(short DayOfWeekOrder)
{
	string arrDayName[7] = {"Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"};
	return arrDayName[DayOfWeekOrder];
}

stDate GetSystemDate()
{
	stDate Date;
	time_t t = time(0);
	tm* now = localtime(&t);

	Date.Year = now->tm_year + 1900;
	Date.Month = now->tm_mon + 1;
	Date.Day = now->tm_mday;

	return Date;
}

bool IsEndOfWeek(stDate Date)
{
	return DayOfWeekOrder(Date) == 6;
}

bool IsWeekEnd(stDate Date)
{
	return (DayOfWeekOrder(Date) == 5 || DayOfWeekOrder(Date) == 6);
}
```

```
bool IsBusinessDay(stDate Date)
{
	return !IsWeekEnd(Date);
}

stDate CalculateVacationReturnDate(stDate DateFrom, short VacationDays)
{
	short WeekEndCounter = 0;
	//in case the data  is weekend keep adding one day util you reach business day
	//we get rid of all weekends before the first business day
	while (IsWeekEnd(DateFrom))
	{
		DateFrom = IncreaseDateByOneDay(DateFrom);
	}
	//here we increase the vacation dates to add all weekends to it.
	for (short i = 1; i <= VacationDays + WeekEndCounter; i++)
	{
		if (IsWeekEnd(DateFrom))
			WeekEndCounter++;

		DateFrom = IncreaseDateByOneDay(DateFrom);
	}
	//in case the return date is weekend keep adding one day until you reach business day
	while (IsWeekEnd(DateFrom))
	{
		DateFrom = IncreaseDateByOneDay(DateFrom);
	}
	return DateFrom;
}

void PrintResult()
{
	cout << "\n\tVacation Starts:" << endl;
	stDate DateFrom = ReadFullDate();

	short VacationDays = 0;
	cout << "\n\tPlease enter Vacation days: ";
	cin >> VacationDays;

	cout << "\n\tReturn Date: ";

	stDate ReturnDate = CalculateVacationReturnDate(DateFrom, VacationDays);

	cout << DayShortName(DayOfWeekOrder(ReturnDate)) << " , ";
	cout << ReturnDate.Day << "/" << ReturnDate.Month << "/" <<
ReturnDate.Year << endl;

}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
}
```

Problem #56 - Is Date 1 After Date 2

Write a program to read Date1 & Date2, and check if Date1 is after Date2 or not.

اكتب برنامجًا لقراءة Date1 وDate2، وتحقق مما إذا كان Date1 بعد Date2 أم لا.

Solving:

```
#pragma warning (disable : 4996)

#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);

	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);
	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}
```

```
Enter Date 1:
Please Enter a day  : 29
Please Enter a month: 5
PLease enter a year : 2024

Enter Date 2:
Please Enter a day  : 28
Please Enter a month: 5
PLease enter a year : 2024

Yes, Date1 after Date2
```

```
bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

short NumberDaysInMonth(short year, short month)
{
	if (month < 1 || month > 12)
		return 0;

	short ArrMon[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
	return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsDat1AfterDate2(stDate Date1, stDate Date2)
{
	return (Date1.Year > Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
		(Date1.Month > Date2.Month ? true : (Date1.Month == Date2.Month ?
			Date1.Day > Date2.Day : false)) : false);
}

void PrintResult()
{
	cout << "\n\tEnter Date 1:" << endl;
	stDate Date1 = ReadFullDate();

	cout << "\n\tEnter Date 2:" << endl;
	stDate Date2 = ReadFullDate();

	if(IsDat1AfterDate2(Date1, Date2))
		cout << "\n\tYes, Date1 after Date2" << endl;
	else
		cout << "\n\tNo, Date1 is not after Date2" << endl;
}

int main()
{
	system("color f0");
	PrintResult();
	system("pause>0");
	return 0;
}
```

==========

Problem #57 - Compare Date Function

Write a program to read Date1 & Date2, and write a function to compare dates, it should return:

اكتب برنامجًا لقراءة Date1 و Date2 ، واكتب دالة لمقارنة التواريخ، فيجب أن يُرجع:

-1 Before

```
Enter Date 1:
Please Enter a day  : 25
Please Enter a month: 5
PLease enter a year : 2024

Enter Date 2:
Please Enter a day  : 29
Please Enter a month: 5
PLease enter a year : 2024

Compare Result = -1
```

0 Equal

```
Enter Date 1:
Please Enter a day  : 29
Please Enter a month: 5
PLease enter a year : 2024

Enter Date 2:
Please Enter a day  : 29
Please Enter a month: 5
PLease enter a year : 2024

Compare Result = 0
```

1 After

```
Enter Date 1:
Please Enter a day  : 29
Please Enter a month: 5
PLease enter a year : 2024

Enter Date 2:
Please Enter a day  : 25
Please Enter a month: 5
PLease enter a year : 2024

Compare Result = 1
```

Solving:

```
#pragma warning (disable : 4996)

#include<iostream>
using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);

	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

enum enCompare { Before = -1, Equal = 0, After = 1 };
```

```
bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
	return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day <
Date2.Day : false)) : false);
}

bool IsDat1EqualToDate2(stDate Date1, stDate Date2)
{
	return ((Date1.Year == Date2.Year) && (Date1.Month == Date2.Month) &&
(Date1.Day == Date2.Day) ? true : false);
}

bool Date1AfterDate2(stDate Date1, stDate Date2)
{
	return (!IsDat1BeforDate2(Date1, Date2)) && (!IsDat1EqualToDate2(Date1,
Date2)) ? true : false;
}

enCompare CompareDates(stDate Date1, stDate Date2)
{
	if (IsDat1BeforDate2(Date1, Date2))
		return enCompare::Before;

	else if (IsDat1EqualToDate2(Date1, Date2))
		return enCompare::Equal;

	/*else if (Date1AfterDate2(Date1, Date2))
		return enCompare::After;*/

		// This is Faster
	return enCompare::After;
}

void PrintDate1AfterDate2Result()
{
	cout << "\n\tEnter Date 1:" << endl;
	stDate Date1 = ReadFullDate();

	cout << "\n\tEnter Date 2:" << endl;
	stDate Date2 = ReadFullDate();

	cout << "\n\tCompare Result = " << CompareDates(Date1, Date2) << endl;

}

int main()
{
	system("color f0");

	PrintDate1AfterDate2Result();

	system("pause>0");
	return 0;
}
```

Problem #58 - Is Overlap Periods

Write a program to read Two Periods and check if they overlap or not?

اكتب برنامجًا لقراءة الفترتين وتأكد من تداخلهما أم لا ؟

Solving:

```
#pragma warning (disable : 4996)

#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

struct stPeriod {
	stDate StartDate;
	stDate EndDate;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);

	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}
```

```
Enter Period 1:

Enter Start Date :
Please Enter a day  : 1
Please Enter a month: 6
PLease enter a year : 2024

Enter End Date :
Please Enter a day  : 10
Please Enter a month: 6
PLease enter a year : 2024

Enter Period 2:

Enter Start Date :
Please Enter a day  : 5
Please Enter a month: 6
PLease enter a year : 2024

Enter End Date :
Please Enter a day  : 20
Please Enter a month: 6
PLease enter a year : 2024

YES, Periods Overlap.
```

```
stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

stPeriod ReadPeriod()
{
	stPeriod Period;
	cout << "\n\tEnter Start Date : \n";
	Period.StartDate = ReadFullDate();

	cout << "\n\tEnter End Date :\n";
	Period.EndDate = ReadFullDate();

	return Period;
}

enum enCompare { Before = -1, Equal = 0, After = 1 };

bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
	return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day <
Date2.Day : false)) : false);
}

bool IsDat1EqualToDate2(stDate Date1, stDate Date2)
{
	return ((Date1.Year == Date2.Year) && (Date1.Month == Date2.Month) &&
(Date1.Day == Date2.Day) ? true : false);
}

bool Date1AfterDate2(stDate Date1, stDate Date2)
{
	return (!IsDat1BeforDate2(Date1, Date2)) && (!IsDat1EqualToDate2(Date1,
Date2)) ? true : false;
}

enCompare CompareDates(stDate Date1, stDate Date2)
{
	if (IsDat1BeforDate2(Date1, Date2))
		return enCompare::Before;

	else if (IsDat1EqualToDate2(Date1, Date2))
		return enCompare::Equal;

	/*else if (Date1AfterDate2(Date1, Date2))
		return enCompare::After;*/

		// This is Faster
	return enCompare::After;
}
```

```
bool IsOverLapPeriod(stPeriod Period1, stPeriod Period2)
{
	if ((CompareDates(Period2.EndDate, Period1.StartDate) ==
enCompare::Before)
		|| (CompareDates(Period2.StartDate, Period1.EndDate) ==
enCompare::After))
	{
		return false;
	}
	else
		return true;
}

void PrintIsOverLapPeriodResult()
{
	cout << "\n\tEnter Period 1:" << endl;
	stPeriod Period1 = ReadPeriod();

	cout << "\n\tEnter Period 2:" << endl;
	stPeriod Period2 = ReadPeriod();

	if(IsOverLapPeriod(Period1, Period2))
		cout << "\n\tYES, Periods Overlap.\n";
	else
		cout << "\n\tNO, Periods do not Overlap.\n";
}

int main()
{
	system("color f0");

	PrintIsOverLapPeriodResult();

	system("pause>0");
	return 0;
}
```

==========

Problem #59 - Period Length In Days

Write a program to read a Period and calculate period length in days.

اكتب برنامج لقراءة فترة وحساب طول الفترة بالأيام.

Solving:

```
#pragma warning (disable : 4996)

#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};
```

```
Enter Period 1:

Enter Start Date:

Please Enter a day  : 25
Please Enter a month: 5
PLease enter a year : 2024

Enter End Date  :

Please Enter a day  : 5
Please Enter a month: 6
PLease enter a year : 2024

Period Length is: 11

Period Length (Including End Date) is: 12
```

```
struct stPeriod {
	stDate StartDate;
	stDate EndDate;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);

	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

stPeriod ReadPeriod()
{
	stPeriod Period;
	cout << "\n\tEnter Start Date: \n\n";
	Period.StartDate = ReadFullDate();

	cout << "\n\tEnter End Date  :\n\n";
	Period.EndDate = ReadFullDate();

	return Period;
}
```

```
bool IsLeapYear(short year)
{
     return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true :
false;
}

short NumberDaysInMonth(short year, short month)
{
     if (month < 1 || month > 12)
          return 0;

     short ArrMon[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

     return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
     return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day <
Date2.Day : false)) : false);
}

bool IsLastDayInMonth(stDate Date)
{
     return Date.Day == NumberDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{
     return (month == 12);
}

stDate IncreaseDateByOneDay(stDate Date)
{
     if (IsLastDayInMonth(Date))
     {
          if (IsLastMonthInYear(Date.Month))
          {
               Date.Day = 1;
               Date.Month = 1;
               Date.Year++;
          }
          else
          {
               Date.Day = 1;
               Date.Month++;
          }
     }
     else
     {
          Date.Day++;
     }
     return Date;
}
```

```
int GetDifferenceInDays(stDate Date1, stDate Date2, bool includeEndDay = false)
{
	short days = 0;

	while (IsDat1BeforDate2(Date1, Date2))
	{
		days++;
		Date1 = IncreaseDateByOneDay(Date1);
	}
	return includeEndDay ? ++days : days;
}

int CalCulatePeriodLengthInDays(stPeriod Period1, bool includeEndDay = false)
{
	return GetDifferenceInDays(Period1.StartDate, Period1.EndDate,
includeEndDay);
}

void PrintCalCulatePeriodLengthInDays()
{
	cout << "\n\tEnter Period 1:" << endl;
	stPeriod Period1 = ReadPeriod();

	cout << "\n\tPeriod Length is: " << CalCulatePeriodLengthInDays(Period1)
<< endl;

	cout << "\n\tPeriod Length (Including End Date) is: " <<
CalCulatePeriodLengthInDays(Period1, true) << endl;
}

int main()
{
	system("color f0");

	PrintCalCulatePeriodLengthInDays();
	system("pause>0");
	return 0;
}
```

==========

Problem #60 - Is Date Within Period?

Write a program to read a Period and Date, then check if date is within this period or not?

أكتب برنامج لقراءة الفترة والتاريخ ثم تأكد إذا كان التاريخ ضمن هذه الفترة أم لا؟

Solving:

```
#pragma warning (disable : 4996)

#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};
```

```
Enter Period 1:
Enter Start Date :

Please Enter a day  : 30
Please Enter a month: 5
PLease enter a year : 2024

Enter End Date :

Please Enter a day  : 10
Please Enter a month: 6
PLease enter a year : 2024

Enter Date to check:

Please Enter a day  : 31
Please Enter a month: 5
PLease enter a year : 2024

YES, Date is within period.
```

```
struct stPeriod {
	stDate StartDate;
	stDate EndDate;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);

	return m;
}

short ReadDay()
{
	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);

	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

stPeriod ReadPeriod()
{
	stPeriod Period;
	cout << "\tEnter Start Date : \n\n";
	Period.StartDate = ReadFullDate();

	cout << "\n\tEnter End Date :\n\n";
	Period.EndDate = ReadFullDate();

	return Period;
}
```

```
enum enCompare { Before = -1, Equal = 0, After = 1 };

bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
     return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day <
Date2.Day : false)) : false);
}

bool IsDat1EqualToDate2(stDate Date1, stDate Date2)
{
     return ((Date1.Year == Date2.Year) && (Date1.Month == Date2.Month) &&
(Date1.Day == Date2.Day) ? true : false);
}

bool Date1AfterDate2(stDate Date1, stDate Date2)
{
     return (!IsDat1BeforDate2(Date1, Date2)) && (!IsDat1EqualToDate2(Date1,
Date2)) ? true : false;
}

enCompare CompareDates(stDate Date1, stDate Date2)
{
     if (IsDat1BeforDate2(Date1, Date2))
          return enCompare::Before;

     else if (IsDat1EqualToDate2(Date1, Date2))
          return enCompare::Equal;

     /*else if (Date1AfterDate2(Date1, Date2))
          return enCompare::After;*/

     return enCompare::After;              // This is Faster
}

bool IsDateInPeriod(stPeriod Period1, stDate Date)
{
     return !((CompareDates(Date, Period1.StartDate) == enCompare::Before)
               ||
               (CompareDates(Date, Period1.EndDate) == enCompare::After));
}

void PrintCalCulatePeriodLengthInDays()
{
     cout << "\n\tEnter Period 1:" << endl;
     stPeriod Period1 = ReadPeriod();
     cout << "\n\tEnter Date to check: \n\n";
     stDate Date = ReadFullDate();
     if (IsDateInPeriod(Period1, Date))
     {
          cout << "\n\tYES, Date is within period.\n";
     }
     else
     {
          cout << "\n\tNO, Date is not within period.\n";
     }
}
```

```
int main()
{
	system("color f0");

	PrintCalCulatePeriodLengthInDays();

	system("pause>0");
	return 0;
}
```

==========

Problem #61 - Count Overlap Days

Write a program to read a Two Periods, then count Overlap days.

اكتب برنامجًا لقراءة الفترتين، ثم قم بعد الأيام المتداخلة.

Solving:

```
#pragma warning (disable : 4996)
#include<iostream>

using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

struct stPeriod {
	stDate StartDate;
	stDate EndDate;
};

short ReadYear()
{	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{	short m;
	do
	{
		cout << "\tPlease Enter a month: ";
		cin >> m;
	} while (m < 1 || m > 12);
	return m;
}

short ReadDay()
{	short d;
	do
	{
		cout << "\tPlease Enter a day  : ";
		cin >> d;
	} while (d < 1 || d > 31);
	return d;
}
```

```
Enter Period 1:
Enter Start Date :

Please Enter a day  : 1
Please Enter a month: 6
PLease enter a year : 2024

Enter End Date :

Please Enter a day  : 10
Please Enter a month: 6
PLease enter a year : 2024

Enter Period 2:
Enter Start Date :

Please Enter a day  : 5
Please Enter a month: 6
PLease enter a year : 2024

Enter End Date :

Please Enter a day  : 31
Please Enter a month: 12
PLease enter a year : 2050

Overlap days count is: 5
```

```
stDate ReadFullDate()
{     stDate Date;
      Date.Day = ReadDay();
      Date.Month = ReadMonth();
      Date.Year = ReadYear();
      return Date;
}

stPeriod ReadPeriod()
{     stPeriod Period;
      cout << "\tEnter Start Date : \n\n";
      Period.StartDate = ReadFullDate();
      cout << "\n\tEnter End Date :\n\n";
      Period.EndDate = ReadFullDate();
      return Period;
}

bool IsLeapYear(short year)
{ return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false; }

short NumberDaysInMonth(short year, short month)
{
      if (month < 1 || month > 12)
            return 0;

      short ArrMon[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
      return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsLastDayInMonth(stDate Date)
{ return Date.Day == NumberDaysInMonth(Date.Year, Date.Month); }

bool IsLastMonthInYear(short month)
{ return (month == 12); }

stDate IncreaseDateByOneDay(stDate Date)
{
      if (IsLastDayInMonth(Date))
      {
            if (IsLastMonthInYear(Date.Month))
            {
                  Date.Day = 1;
                  Date.Month = 1;
                  Date.Year++;
            }
            else
            {
                  Date.Day = 1;
                  Date.Month++;
            }
      }
      else
      {
            Date.Day++;
      }
      return Date;
}
```

```
enum enCompare { Before = -1, Equal = 0, After = 1 };

bool IsDat1BeforDate2(stDate Date1, stDate Date2)
{
     return (Date1.Year < Date2.Year) ? true : ((Date1.Year == Date2.Year) ?
(Date1.Month < Date2.Month ? true : (Date1.Month == Date2.Month ? Date1.Day <
Date2.Day : false)) : false);
}

bool IsDat1EqualToDate2(stDate Date1, stDate Date2)
{
     return ((Date1.Year == Date2.Year) && (Date1.Month == Date2.Month) &&
(Date1.Day == Date2.Day) ? true : false);
}

bool Date1AfterDate2(stDate Date1, stDate Date2)
{
     return (!IsDat1BeforDate2(Date1, Date2)) && (!IsDat1EqualToDate2(Date1,
Date2)) ? true : false;
}

enCompare CompareDates(stDate Date1, stDate Date2)
{
     if (IsDat1BeforDate2(Date1, Date2))
          return enCompare::Before;

     else if (IsDat1EqualToDate2(Date1, Date2))
          return enCompare::Equal;

     /*else if (Date1AfterDate2(Date1, Date2))
          return enCompare::After;*/

          // This is Faster
     return enCompare::After;
}

bool IsOverLapPeriod(stPeriod Period1, stPeriod Period2)
{
     if ((CompareDates(Period2.EndDate, Period1.StartDate) == enCompare::Before)
          || (CompareDates(Period2.StartDate, Period1.EndDate) == enCompare::After))
     {
          return false;
     }
     else
          return true;
}

int GetDifferenceInDays(stDate Date1, stDate Date2, bool includeEndDay = false)
{
     short days = 0;
     while (IsDat1BeforDate2(Date1, Date2))
     {
          days++;
          Date1 = IncreaseDateByOneDay(Date1);
     }
     return includeEndDay ? ++days : days;
}
```

```
int PeriodLengthInDays(stPeriod Period1, bool includeEndDay = false)
{
	return GetDifferenceInDays(Period1.StartDate, Period1.EndDate, includeEndDay);
}

bool IsDateInPeriod(stPeriod Period1, stDate Date)
{
	return !((CompareDates(Date, Period1.StartDate) == enCompare::Before)
		||
		(CompareDates(Date, Period1.EndDate) == enCompare::After));
}

int CountOverlapDays(stPeriod Period1, stPeriod Period2)
{
	int Period1Length = PeriodLengthInDays(Period1, true);
	int Period2Length = PeriodLengthInDays(Period2, true);
	int OverlapDays = 0;
	if (!IsOverLapPeriod(Period1, Period2))
		return 0;
	if(Period1Length < Period2Length)
	{
		while (IsDat1BeforDate2(Period1.StartDate, Period1.EndDate))
		{
			if(IsDateInPeriod(Period2, Period1.StartDate))
				OverlapDays++;

			Period1.StartDate = IncreaseDateByOneDay(Period1.StartDate);
		}
	}
	else
	{
		while (IsDat1BeforDate2(Period2.StartDate, Period2.EndDate))
		{
			if (IsDateInPeriod(Period1, Period2.StartDate))
				OverlapDays++;
			Period2.StartDate = IncreaseDateByOneDay(Period2.StartDate);
		}
	}
	return OverlapDays;
}

void PrintIsOverLapPeriodResult()
{
	cout << "\n\tEnter Period 1:" << endl;
	stPeriod Period1 = ReadPeriod();
	cout << "\n\tEnter Period 2:" << endl;
	stPeriod Period2 = ReadPeriod();
	cout << "\n\tOverlap days count is: " << CountOverlapDays(Period1, Period2) << endl;
}

int main()
{
	system("color f0");

	PrintIsOverLapPeriodResult();
	system("pause>0");
	return 0;
}
```

Problem #62 - Validate Date

Write a program to read Date and write a function to validate this date.

اكتب برنامجًا لقراءة التاريخ واكتب دالة للتحقق من صحة هذا التاريخ.

```
Enter Date:

Please Enter a day  : 31
Please Enter a month: 4
PLease enter a year : 2024

NO, Date is a NOT valide date.
```

```
Enter Date:

Please Enter a day  : 29
Please Enter a month: 2
PLease enter a year : 2023

NO, Date is a NOT valide date.
```

```
Enter Date:

Please Enter a day  : 29
Please Enter a month: 2
PLease enter a year : 2024

YES, Date is a valide date.
```

```
Enter Date:

Please Enter a day  : 35
Please Enter a month: 5
PLease enter a year : 2024

NO, Date is a NOT valide date.
```

```
Enter Date:

Please Enter a day  : 25
Please Enter a month: 15
PLease enter a year : 2024

NO, Date is a NOT valide date.
```

My Solving:

```cpp
#include<iostream>
using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{	short m;
	cout << "\tPlease Enter a month: ";
	cin >> m;
	return m;
}

short ReadDay()
{	short d;
	cout << "\tPlease Enter a day  : ";
	cin >> d;
	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}
```

```
bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

short NumberDaysInMonth(short year, short month)
{
	if (month < 1 || month > 12)
		return 0;

	short ArrMon[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

	return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}

bool IsLastDayInMonth(stDate Date)
{
	return Date.Day == NumberDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{
	return (month == 12);
}

bool IsValideDate(stDate Date)
{
	return Date.Day > 0 && Date.Day <= (NumberDaysInMonth(Date.Year, Date.Month));
}

void PrintIsValideDate()
{
	cout << "\n\tEnter Date: \n\n";
	stDate Date = ReadFullDate();

	if (IsValideDate(Date))
		cout << "\n\tYES, Date is a valide date.\n";
	else
		cout << "\n\tNO, Date is a NOT valide date.\n";
}

int main()
{
	system("color f0");

	PrintIsValideDate();

	system("pause>0");
	return 0;
}
```

Abu-Hadhoud Solving:

```
#include<iostream>
using namespace std;

struct stDate {
	short Year;
	short Month;
	short Day;
};

short ReadYear()
{
	short y;
	cout << "\tPLease enter a year : ";
	cin >> y;
	return y;
}

short ReadMonth()
{
	short m;
	cout << "\tPlease Enter a month: ";
	cin >> m;
	return m;
}

short ReadDay()
{
	short d;
	cout << "\tPlease Enter a day  : ";
	cin >> d;
	return d;
}

stDate ReadFullDate()
{
	stDate Date;
	Date.Day = ReadDay();
	Date.Month = ReadMonth();
	Date.Year = ReadYear();
	return Date;
}

bool IsLeapYear(short year)
{
	return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0) ? true : false;
}

short NumberDaysInMonth(short year, short month)
{
	if (month < 1 || month > 12)
		return 0;

	short ArrMon[12] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

	return (month == 2) ? (IsLeapYear(year) ? 29 : 28) : ArrMon[month - 1];
}
```

```
bool IsLastDayInMonth(stDate Date)
{
	return Date.Day == NumberDaysInMonth(Date.Year, Date.Month);
}

bool IsLastMonthInYear(short month)
{
	return (month == 12);
}

bool IsValideDate(stDate Date)
{
	if (Date.Day < 1 || Date.Day > 31)
		return false;

	if (Date.Month < 1 || Date.Month > 12)
		return false;

	if (Date.Month == 2)
	{
		if (IsLeapYear(Date.Year))
		{
			if (Date.Month > 29)
				return false;
		}
		else
		{
			if (Date.Month > 28)
				return false;
		}
	}

	short DaysInMonth = (NumberDaysInMonth(Date.Year, Date.Month));
	if (Date.Day > DaysInMonth)
		return false;

	return true;
}

void PrintIsValideDate()
{
	cout << "\n\tEnter Date: \n\n";
	stDate Date = ReadFullDate();

	if (IsValideDate(Date))
		cout << "\n\tYES, Date is a valide date.\n";
	else
		cout << "\n\tNO, Date is a NOT valide date.\n";
}

int main()
{
	system("color f0");
	PrintIsValideDate();
	system("pause>0");
	return 0;
}
```

Problem #63 - #64 - Convert String To Date & Convert Date To String

Write a program to: **اكتب برنامجًا لـ:**

- **Read Date String.** - قراءة سلسلة التاريخ.
- **Convert it to date structure.** - تحويله إلى هيكل التاريخ.
- **Print Day, Month, Year separately.** - طباعة اليوم والشهر والسنة بشكل منفصل.
- **Then convert Date Structure to string and print it on the screen.**
- ثم قم بتحويل بنية التاريخ إلى سلسلة وطباعتها على الشاشة.

Note: Write the following functions: **ملاحظة: اكتب الوظائف التالية:**

- **Function: StringToDate.** - دالة: سلسلة الى تاريخ.
- **Function: DateToString.** - دالة: تاريخ إلى سلسلة.

```
Please enter date dd/mm/yyyy: 30/6/2024

Convert String (30/6/2024) To Date:
-----------------------------------
Day  : 30
Month: 6
Year : 2024

Convert Date To String:
-----------------------
You entered : 30/6/2024
```

Solving:

```
#include<iostream>
#include<string>
#include<vector>
using namespace std;

string ReadStringDate(string Message)
{
	string StringDate;
	cout << Message;
	getline(cin >> ws, StringDate);
	return StringDate;
}

struct stDate {
	short Year;
	short Month;
	short Day;
};

vector<string>SplitString(string S1, string Delimi)
{
	vector<string>vString;

	short pos = 0;
	string sWord;

	while ((pos = S1.find(Delimi)) != std::string::npos)
	{
		sWord = S1.substr(0, pos);
```

```
            if (sWord != "")
            {
                  vString.push_back(sWord);
            }
            S1.erase(0, pos + Delimi.length());
      }
      if (S1 != " ")
      {
            vString.push_back(S1);
      }
      return vString;
}

stDate StringToDate(string SD)
{
      vector<string>vString;
      stDate Date;

      vString = SplitString(SD, "/");

      Date.Day = stoi(vString[0]);
      Date.Month = stoi(vString[1]);
      Date.Year = stoi(vString[2]);
      return Date;
}

string DateToString(stDate Date, string Delimi)
{
      return to_string(Date.Day) + Delimi + to_string(Date.Month) + Delimi +
to_string(Date.Year);
}

void PrintResult()
{
      string SD = ReadStringDate("\n\tPlease enter date dd/mm/yyyy: ");
      stDate Date;
      Date = StringToDate(SD);
      cout << "\n\tConvert String (" << SD << ") To Date: \n";
      cout << "\t----------------------------------\n";
      cout << "\tDay  : " << Date.Day << endl;
      cout << "\tMonth: " << Date.Month << endl;
      cout << "\tYear : " << Date.Year << endl;
      cout << "\n\tConvert Date To String: \n";
      cout << "\t----------------------\n";
      cout << "\tYou entered : " << DateToString(Date, "/") << endl;
}

int main()
{
      system("color f0");

      PrintResult();

      system("pause>0");
      return 0;
}
```

Problem #65 - Format Date

Write a program to Read Date and write a function to format that date.

اكتب برنامجًا لقراءة التاريخ واكتب دالة لتنسيق ذلك التاريخ.

```
Please enter date dd/mm/yyyy: 31/12/2024
----------------------------------------
31/12/2024

2024/12/31

12/31/2024

31-12-2024

12-31-2024

Day: 31   Month: 12   Year: 2024
```

Solving:

```
#include<iostream>
#include<string>
#include<vector>
using namespace std;

string ReadStringDate(string Message)
{
	string StringDate;
	cout << Message;
	getline(cin >> ws, StringDate);
	return StringDate;
}

struct stDate {
	short Year;
	short Month;
	short Day;
};

vector<string>SplitString(string S1, string Delimi)
{
	vector<string>vString;
	short pos = 0;
	string sWord;
	while ((pos = S1.find(Delimi)) != std::string::npos)
	{
		sWord = S1.substr(0, pos);

		if (sWord != "")
		{
			vString.push_back(sWord);
		}
		S1.erase(0, pos + Delimi.length());
	}
	if (S1 != " ")
	{
		vString.push_back(S1);
	}
	return vString;
}

stDate StringToDate(string SD)
{
	vector<string>vString;
	stDate Date;
	vString = SplitString(SD, "/");
	Date.Day = stoi(vString[0]);
	Date.Month = stoi(vString[1]);
	Date.Year = stoi(vString[2]);
	return Date;
}
```

```
string ReplaceWordsInString(string S1, string Word1, string Word2)
{
	short pos = S1.find(Word1);

	while (pos != std::string::npos)
	{
		S1 = S1.replace(pos, Word1.length(), Word2);
		pos = S1.find(Word1); // find next word
	}
	return S1;
}

string FormateDate(stDate Date, string DateFormat = "dd/mm/yyyy")
{
	string FormattedDateString = "";
	FormattedDateString = ReplaceWordsInString(DateFormat, "dd",
to_string(Date.Day));
	FormattedDateString = ReplaceWordsInString(FormattedDateString, "mm",
to_string(Date.Month));
	FormattedDateString = ReplaceWordsInString(FormattedDateString, "yyyy",
to_string(Date.Year));
	return FormattedDateString;
}

void PrintResult()
{
	string SD = ReadStringDate("\n\tPlease enter date dd/mm/yyyy: ");

	stDate Date;
	Date = StringToDate(SD);

	cout << "\t----------------------------------------\n";
	cout << "\t" << FormateDate(Date) << "\n";
	cout << "\n\t" << FormateDate(Date, "yyyy/mm/dd") << "\n";
	cout << "\n\t" << FormateDate(Date, "mm/dd/yyyy") << "\n";
	cout << "\n\t" << FormateDate(Date, "dd-mm-yyyy") << "\n";
	cout << "\n\t" << FormateDate(Date, "mm-dd-yyyy") << "\n";
	cout << "\n\t" << FormateDate(Date, "Day: dd   Month: mm   Year: yyyy") <<
"\n";
}

int main()
{
	system("color f0");

	PrintResult();

	system("pause>0");
	return 0;
}
```

Project 1 Bank Extension 2 Requirements

```
==============================
        Login Screen
==============================


Please Enter Username: Admin
Please Enter Password: 2344
```

```
==============================
        Login Screen
==============================

Invalid Username/Password!

Please Enter Username:
```

```
==============================
        Login Screen
==============================

Invalid Username/Password!

Please Enter Username: Admin
Please Enter Password: 1234
```

```
=====================================================
                Main Menue Screen
=====================================================
        [1] Show Client List.
        [2] Add New Client.
        [3] Delete Client.
        [4] Update Client Info.
        [5] Find Client.
        [6] Transactions.
        [7] Manage Users.        Add new item
        [8] Logout.              Changed Exit to Logout
=====================================================
Choose what do you want to do ? [1 to 8] ?
```

```
=====================================================
            Manage Users Menu Screen
=====================================================
        [1] List Users.
        [2] Add New User.
        [3] Delete User.
        [4] Update User.
        [5] Find User.
        [6] Main Menu.
=====================================================
Choose what do you want to do ? [1 to 6] ?
```

```
=====================================================
            Manage Users Menu Screen
=====================================================
        [1] List Users.
        [2] Add New User.
        [3] Delete User.
        [4] Update User.
        [5] Find User.
        [6] Main Menu.
=====================================================
Choose what do you want to do ? [1 to 6] ? 1
```

```
                    Users List (2) User(s)

=============================================================================
|User Name          |Password                 |Permissions
=============================================================================
|Admin              |1234                     |-1
|User2              |4321                     |17

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

Press any key to go back to Manage Users Menue...
```

```
=====================================================
            Manage Users Menu Screen
=====================================================
        [1] List Users.
        [2] Add New User.
        [3] Delete User.
        [4] Update User.
        [5] Find User.
        [6] Main Menu.
=====================================================
Choose what do you want to do ? [1 to 6] ? 2
```

```
=====================================================
            Add New Users Screen
=====================================================

Adding New User:

Enter Username: User2
User with [User2] already exists, Enter Another Username:
```

```
=====================================================
            Add New Users Screen
=====================================================

Adding New User:

Enter Username: User2
User with [User2] already exists, Enter Another Username: User3
Enter Password: 7539
Do you want to give full access? (Y/N) :
```

```
=====================================================
            Add New Users Screen
=====================================================

Adding New User:

Enter Username: User2
User with [User2] already exists, Enter Another Username: User3
Enter Password: 7539
Do you want to give full access? (Y/N) : y

User Added Successfully, do you want to add more Users? (Y/N):
```

```
=====================================================
            Add New Users Screen
=====================================================

Adding New User:

Enter Username: User4
Enter Password: 95123
Do you want to give full access? (Y/N) : n

Do you want to give access to:
Show Client List ?(Y/N): y
Add New Client ?(Y/N): n
Delete Client ?(Y/N): n
Update Client ?(Y/N): n
Find Client ?(Y/N): y
Transactions ?(Y/N): n
Manage Users ?(Y/N): n

User Added Successfully, do you want to add more Users? (Y/N):
```

```
=====================================================
            Add New Users Screen
=====================================================

Adding New User:

Enter Username: User4
Enter Password: 95123
Do you want to give full access? (Y/N) : n

Do you want to give access to:
Show Client List ?(Y/N): y
Add New Client ?(Y/N): n
Delete Client ?(Y/N): n
Update Client ?(Y/N): n
Find Client ?(Y/N): y
Transactions ?(Y/N): n
Manage Users ?(Y/N): n

User Added Successfully, do you want to add more Users? (Y/N): n


Press any key to go back to Manage Users Menue...
```

```
============================================================
                Manage Users Menu Screen
============================================================
        [1] List Users.
        [2] Add New User.
        [3] Delete User.
        [4] Update User.
        [5] Find User.
        [6] Main Menu.
============================================================
Choose what do you want to do ? [1 to 6] ? 3
```

```
================================================
            Delete Users Screen
================================================

Please enter Username: User3

The Following are the User details:

Username    : User3
Password    : 7539
Permissions : -1

Are you sure you want delete this User ? (Y / N) :
```

```
================================================
            Delete Users Screen
================================================

Please enter Username: User3

The Following are the User details:

Username    : User3
Password    : 7539
Permissions : -1

Are you sure you want delete this User ? (Y / N) : y


User Deleted Successfully.


Press any key to go back to Manage Users Menue...
```

```
============================================================
                Manage Users Menu Screen
============================================================
        [1] List Users.
        [2] Add New User.
        [3] Delete User.
        [4] Update User.
        [5] Find User.
        [6] Main Menu.
============================================================
Choose what do you want to do ? [1 to 6] ? 4
```

```
============================================================
                Update Users Screen
============================================================

Please enter Username: User3


User with Username (User3) is not found!


Press any key to go back to Manage Users Menue...
```

```
================================================
            Update Users Screen
================================================

Please enter Username: User2

The Following are the User details:

Username    : User2
Password    : 4321
Permissions : 17

Are you sure you want update this User ? (Y / N) :
```

```
================================================
            Update Users Screen
================================================

Please enter Username: User2

The Following are the User details:

Username    : User2
Password    : 4321
Permissions : 17

Are you sure you want update this User ? (Y / N) : y
Enter Password ? 78965
Do you want to give full access? (Y/N) : n

Do you want to give access to:
Show Client List ?(Y/N): y
Add New Client ?(Y/N): n
Delete Client ?(Y/N): n
Update Client ?(Y/N): n
Find Client ?(Y/N): n
Transactions ?(Y/N): n
Manage Users ?(Y/N): n


User Updated Successfully.


Press any key to go back to Manage Users Menue...
```

```
============================================================
                Manage Users Menu Screen
============================================================
        [1] List Users.
        [2] Add New User.
        [3] Delete User.
        [4] Update User.
        [5] Find User.
        [6] Main Menu.
============================================================
Choose what do you want to do ? [1 to 6] ? 5
```

```
============================================================
        Find User Screen
============================================================

Please enter Username: User3


User with Username (User3) not found!


Press any key to go back to Manage Users Menue...
```

```
============================================================
        Find User Screen
============================================================

Please enter Username: User2

The Following are the User details:

Username    : User2
Password    : 78965
Permissions : 1


Press any key to go back to Manage Users Menue...
```

Solving:

```
#include<iostream>
#include<string>
#include<fstream>
#include<vector>
#include<iomanip>

using namespace std;

//-------------------------------------------------------------------//
//---------------------------- 7- Project 1 -------------------------//
//-------------------------------------------------------------------//

// عمل متغير ثابت لاسم الملف في البداية
const string ClientsFileName = "Clients.txt";
const string UsersFileName = "Users.txt";

void ShowMainMenueScreen();
void ShowTransactionsMenuScreen();
void ShowManageUsersMenuScreen();
void ShowAccessDeniedMessage();
void GoBackToMainMenue();

struct stClient {
	string Account_Number;
	string PinCode;
	string Full_Name;
	string Phone;
	float AccountBalance;

	bool MarkForDelete = false;		// Flag - لا يخزن على الفايل.. لكن نستخدمه عند وضع قيمة ترو له كعلامة للحذف
};

struct stUser {
	string Username;
	string Password;
	short Permissions;

	bool MarkForDelete = false;		// Flag - لا يخزن على الفايل.. لكن نستخدمه عند وضع قيمة ترو له كعلامة للحذف
};

stUser CurrentUser;

stClient CovertLineToRecord_Clients(string line);

enum enMainMenuPermissions {
	eAll = -1, pClientList = 1, pAddNewClien = 2, pDeleteClient = 4,
pUpdateClient = 8,
	pFindClient = 16, pTransactions = 32, pManageUsers = 64
};

bool CheckAccessPermission(enMainMenuPermissions Permission);
```

```
//------------------- Add Clients To File ----------------------//

bool ClientExistsByAccountNumber(string AccountNumber, string filename)
                                          // للتحقق إذا كان الـ Account Number موجود أم لا
{
      vector<stClient>vClient;
      fstream NewFile;
      NewFile.open(filename, ios::in);
      if (NewFile.is_open())
      {
            string line;
            stClient Client;
            while (getline(NewFile, line))
            {
                  Client = CovertLineToRecord_Clients(line);
                  if (Client.Account_Number == AccountNumber)
                  {
                        NewFile.close();// مهم جداً ينقفل الملف هنا
                        return true;
                  }
                  vClient.push_back(Client);
            }
            NewFile.close();
      }
      return false;
}

stClient ReadClientData()
{
      stClient Client;
      cout << "Enter Account Number ? ";
      // usage for std::ws will extract all the whitespace character
                              //سيؤدي استخدام std::ws إلى استخراج كل أحرف المسافة البيضاء
      getline(cin >> ws, Client.Account_Number);

                                          //للتحقق إذا كان الـ Account Number موجود أم لا
      while (ClientExistsByAccountNumber(Client.Account_Number, ClientsFileName))
      {
            cout << "Client with [" << Client.Account_Number << "] already
exists, Enter Another Account Number: ";
            getline(cin >> ws, Client.Account_Number);
      }

      cout << "Enter PinCode ? ";
      getline(cin, Client.PinCode);

      cout << "Enter Ful Name ? ";
      getline(cin, Client.Full_Name);

      cout << "Enter Phone ? ";
      getline(cin, Client.Phone);

      cout << "Enter Account Balance ? ";
      cin >> Client.AccountBalance;

      return Client;
}
```

```
string ConvertRecordToLine_Clients(stClient sClient, string Delimi = "#//#")
{
	string line = "";

	line += sClient.Account_Number + Delimi;
	line += sClient.PinCode + Delimi;
	line += sClient.Full_Name + Delimi;
	line += sClient.Phone + Delimi;
	line += to_string(sClient.AccountBalance);
	return line;
}

void AddClientDataToFile(string filename, string Line)
{
	fstream to_File;

	to_File.open(filename, ios::out | ios::app);

	if (to_File.is_open())
	{
		to_File << Line << endl;
		to_File.close();
	}
}

void AddNewClient()
{
	stClient Client;
	Client = ReadClientData();
	AddClientDataToFile(ClientsFileName, ConvertRecordToLine_Clients(Client));
}

void AddClients()
{
	char AddMore = 'Y';
	do
	{
		cout << "\nAdding New Client:\n\n";
		AddNewClient();
		cout << "\nClient Added Successfully, do you want to add more clients? (Y/N): ";
		cin >> AddMore;
	} while (toupper(AddMore) == 'Y');
}

void ShowAddNewClientsScreen()
{
	if (!CheckAccessPermission(enMainMenuPermissions::pAddNewClien))
	{
		ShowAccessDeniedMessage();
		return;
	}
	system("cls");
	cout << "\n=====================================================\n";
	cout << "\t\tAdd New Clients Screen\n";
	cout << "=====================================================\n";
	AddClients();
}
```

```
//-------------------- Show All Clients ----------------------//

void HeaderOfTable(vector<stClient>vClient)
{
	cout << "\n\t\t\t\tClient List (" << vClient.size() << ") Client(s)\n";
	cout << "\n=========================================";
	cout << "=========================================\n";
	cout << "|" << left << setw(15) << "Account Number";
	cout << "|" << left << setw(12) << "Pic Code";
	cout << "|" << left << setw(30) << "Client Name";
	cout << "|" << left << setw(14) << "Phone";
	cout << "|" << left << setw(14) << "Balance";
	cout << "\n=========================================";
	cout << "=========================================\n";
}

vector<string> SplitString(string line, string Delimi = "#//#")
{
	vector<string>vString;
	short pos = 0;
	string sWord = "";

	while ((pos = line.find(Delimi)) != std::string::npos)
	{
		sWord = line.substr(0, pos);
		if (sWord != "")
		{
			vString.push_back(sWord);
		}
		line.erase(0, pos + Delimi.length());
	}
	if (line != "")
	{
		vString.push_back(line);
	}
	return vString;
}

stClient CovertLineToRecord_Clients(string line)
{
	vector<string>vString;
	vString = SplitString(line);

	stClient Client;

	Client.Account_Number = vString[0];
	Client.PinCode = vString[1];
	Client.Full_Name = vString[2];
	Client.Phone = vString[3];
	Client.AccountBalance = stof(vString[4]);

	return Client;
}
```

```
vector<stClient> LoadClientDataFromFile(string filename)
{
	vector<stClient>vClient;

	fstream NewFile;

	NewFile.open(filename, ios::in);

	if (NewFile.is_open())
	{
		string line;
		stClient Client;

		while (getline(NewFile, line))
		{
			Client = CovertLineToRecord_Clients(line);
			vClient.push_back(Client);
		}
		NewFile.close();
	}
	return vClient;
}

void MidOfTable(stClient Client)
{
	cout << "|" << left << setw(15) << Client.Account_Number;
	cout << "|" << left << setw(12) << Client.PinCode;
	cout << "|" << left << setw(30) << Client.Full_Name;
	cout << "|" << left << setw(14) << Client.Phone;
	cout << "|" << left << setw(14) << Client.AccountBalance;
}

void PrintMidOfTable(vector<stClient>vClient)
{
	if (vClient.size() == 0)
	{
		cout << "\t\t\tNo Clients Available In The System!";
	}
	else
	{
		for (stClient C : vClient)
		{
			MidOfTable(C);
			cout << endl;
		}
	}
}

void FooterOfTable()
{
	cout << "_________________________________________";
	cout << "_________________________________________\n";
}
```

```
void ShowAllClientsScreen()
{
	if (!CheckAccessPermission(enMainMenuPermissions::pClientList))
	{
		ShowAccessDeniedMessage();
		return;
	}
	vector<stClient>vClient;
	vClient = LoadClientDataFromFile(ClientsFileName);

	HeaderOfTable(vClient);

	PrintMidOfTable(vClient);

	FooterOfTable();
}

//---------------- Find Client By Account Number --------------//

string EnterAccountNumber()
{
	string AccountNumber;
	cout << "Please enter Account Number: ";
	cin >> AccountNumber;
	return AccountNumber;
}

void ReadClientCard(stClient Client)
{
	cout << "\nThe Following are the client details:\n\n";

	cout << "Account Number : " << Client.Account_Number << endl;
	cout << "Pin Code       : " << Client.PinCode << endl;
	cout << "Full Name      : " << Client.Full_Name << endl;
	cout << "Phone          : " << Client.Phone << endl;
	cout << "Account Balance: " << Client.AccountBalance << endl;
}

bool FindClientByAccountNumber1(stClient& Client, string AccountNumber)
{
	vector<stClient>vClient;

	vClient = LoadClientDataFromFile(ClientsFileName);

	for (stClient C : vClient)
	{
		if (C.Account_Number == AccountNumber)
		{
			Client = C;
			return true;
		}
	}
	return false;
}
```

```
void ShowFindClientScreen()
{
	if (!CheckAccessPermission(enMainMenuPermissions::pFindClient))
	{
		ShowAccessDeniedMessage();
		return;
	}
	cout << "\n-----------------------------------------------------\n";
	cout << "\t\Find Client Screen\n";
	cout << "-----------------------------------------------------\n\n";

	string AccountNumber = EnterAccountNumber();
	stClient Client;

	if (FindClientByAccountNumber1(Client, AccountNumber))
	{
		ReadClientCard(Client);
	}
	else
	{
		cout << "\n\nClient with account number (" << AccountNumber << ")
			 not found!\n";
	}
}

//----------- Delete Client By Account Number ---------------//

bool FindClientByAccountNumber2(string AccountNumber, vector<stClient>&
vClient, stClient& Client)
{
	for (stClient& C : vClient)
	{
		if (C.Account_Number == AccountNumber)
		{
			Client = C;
			return true;
		}
	}
	return false;
}

bool MarkClientForDeleteByAccountNumber(string AccountNumber, vector<stClient>&
vClient)
{
	for (stClient& C : vClient)
	{
		if (C.Account_Number == AccountNumber)
		{
			C.MarkForDelete = true;
			return true;
		}
	}
	return false;
}
```

```
vector<stClient> SaveClientDataToFile2(string filename, vector<stClient>& vClient)
{
	fstream NewFile;
	NewFile.open(filename, ios::out); // OverWrite
	string line;

	if (NewFile.is_open())
	{
		for (stClient C : vClient)
		{
			if (C.MarkForDelete == false)
			{
				// We only write records that are not marked for delete.

				line = ConvertRecordToLine_Clients(C);

				NewFile << line << endl;
			}
		}
		NewFile.close();
	}
	return vClient;
}

bool DeleteClientByAccountNumber2(string AccountNumber, vector<stClient>&
vClient)
{
	stClient Client;

	char Answer = 'n';

	if (FindClientByAccountNumber2(AccountNumber, vClient, Client))
	{
		ReadClientCard(Client);
		cout << "\nAre you sure you want delete this client ? (Y / N) : ";
		cin >> Answer;

		if (toupper(Answer) == 'Y')
		{
			MarkClientForDeleteByAccountNumber(AccountNumber, vClient);

			SaveClientDataToFile2(ClientsFileName, vClient);

			// ReFresh Clients
			vClient = LoadClientDataFromFile(ClientsFileName);

			cout << "\n\nClient Deleted Successfully.\n";
			return true;
		}
	}
	else
	{
		cout << "\n\nClient with account number (" << AccountNumber << ") is
not found!\n";
		return false;
	}
}
```

```
void ShowDeleteClientScreen()
{
	if (!CheckAccessPermission(enMainMenuPermissions::pDeleteClient))
	{
		ShowAccessDeniedMessage();
		return;
	}
	cout << "\n-----------------------------------------------------\n";
	cout << "\t\tDelete Client Screen\n";
	cout << "-----------------------------------------------------\n\n";

	vector<stClient> vClient;
	vClient = LoadClientDataFromFile(ClientsFileName);

	string AccountNumber = EnterAccountNumber();
	DeleteClientByAccountNumber2(AccountNumber, vClient);
}

//----------- Update Client By Account Number --------------//

stClient ChangeClientRecord(string AccountNumber)
{
	stClient Client;

	Client.Account_Number = AccountNumber;

	cout << "Enter PinCode ? ";
	getline(cin >> ws, Client.PinCode);

	cout << "Enter Ful Name ? ";
	getline(cin, Client.Full_Name);

	cout << "Enter Phone ? ";
	getline(cin, Client.Phone);

	cout << "Enter Account Balance ? ";
	cin >> Client.AccountBalance;

	return Client;
}

bool UpdateClientByAccountNumber2(string AccountNumber, vector<stClient>&
vClient)
{
	stClient Client;

	char Answer = 'n';

	if (FindClientByAccountNumber2(AccountNumber, vClient, Client))
	{
		ReadClientCard(Client);

		cout << "\nAre you sure you want update this client ? (Y / N) : ";
		cin >> Answer;

		if (toupper(Answer) == 'Y')
		{
```

```
                for (stClient& C : vClient)
                {
                      if (C.Account_Number == AccountNumber)
                      {
                            C = ChangeClientRecord(AccountNumber);
                            break;
                /*طالما انا لاقيت الكلاينت وعدلت معلوماته بعمل بريك علطول مفيش داعي اكمل - اعد على الفاضي
                                                افرض عندي عشرتلاف كلاينت موجودين عندي في الفايل
                                                بدي امشي على العشرتلاف
                                    ما انا اول واحد لاقيته خلاص عدلت عليه اعمل بريك .. البريك هاي تطلعني من اللوب
                ليش عملنا ابريك معلمتش ريتيرن = لان بدي اكمل شغل جوه الفانكشن - الريترن تطلع بره الفانكشن
                                                */
                      }
                }

                SaveClientDataToFile2(ClientsFileName, vClient);

                cout << "\n\nClient Updated Successfully.\n";
                return true;
          }
     }
     else
     {
          cout << "\n\nClient with account number (" << AccountNumber << ") is
not found!\n";
          return false;
     }
}

void ShowUpdateClientScreen()
{
     if (!CheckAccessPermission(enMainMenuPermissions::pUpdateClient))
     {
          ShowAccessDeniedMessage();
          return;
     }
     system("cls");

     cout << "\n-----------------------------------------------------\n";
     cout << "\t\tUpdate Client Info Screen\n";
     cout << "-----------------------------------------------------\n\n";

     string AccountNumber = EnterAccountNumber();

     vector<stClient> vClient;
     vClient = LoadClientDataFromFile(ClientsFileName);

     UpdateClientByAccountNumber2(AccountNumber, vClient);
}

//-------------------------------------------------------------------//
//--------------- 7- Project 2 : Bank Extension 1 -------------------//
//-------------------------------------------------------------------//
```

```
bool BalanceByAccountNumber(string AccountNumber, float Amount,
vector<stClient>vClient)
{
	char Answer = 'Y';

	cout << "\nAre you sure you want perfrom this transaction: (Y/N) ? ";
	cin >> Answer;

	if (toupper(Answer) == 'Y')
	{
		for (stClient& C : vClient)
		{
			if (C.Account_Number == AccountNumber)
			{
				C.AccountBalance += Amount;

				SaveClientDataToFile2(ClientsFileName, vClient);

				cout << "\n\nDone Successfully, New balance is: " <<
C.AccountBalance << endl;

				return true;
			}
		}
		return false;
	}
}

void DepositSrceen()
{
	float DepositAmount = 0;		// DepositAmount = قيمة الإيداع
	char Answer = 'Y';

	vector<stClient>vClient;

	string AccountNumber = EnterAccountNumber();

	vClient = LoadClientDataFromFile(ClientsFileName);

	stClient Client;

	while (!FindClientByAccountNumber2(AccountNumber, vClient, Client))
	{
		cout << "\nClient with [" << AccountNumber << "] does not
exist!\n\n";
		AccountNumber = EnterAccountNumber();
	}

	ReadClientCard(Client);

	cout << "\nPlease enter deposit amount: ";
	cin >> DepositAmount;

	BalanceByAccountNumber(AccountNumber, DepositAmount, vClient);
}
```

```
void WithdrawSrceen()
{
	float WithdrawAmount = 0;		// WithdrawAmount = قيمة السحب
	char Answer = 'Y';

	vector<stClient>vClient;

	string AccountNumber = EnterAccountNumber();

	vClient = LoadClientDataFromFile(ClientsFileName);

	stClient Client;

	while (!FindClientByAccountNumber2(AccountNumber, vClient, Client))
	{
		cout << "\nClient with [" << AccountNumber << "] does not
exist!\n\n";
		AccountNumber = EnterAccountNumber();
	}

	ReadClientCard(Client);

	cout << "\nPlease enter withdraw amount: ";
	cin >> WithdrawAmount;


	while (WithdrawAmount > Client.AccountBalance)
	{
		cout << "Amount exceeds the balance, you can withdraw up to : " <<
Client.AccountBalance << endl;
		cout << "\nPlease enter withdraw amount: ";
		cin >> WithdrawAmount;
	}

	BalanceByAccountNumber(AccountNumber, WithdrawAmount * -1, vClient);
//الضرب في سالب واحد يعطينا النتيجة سالب
}

void HeaderOfBalanceTable(vector<stClient>vClient)
{
	cout << "\n\t\t\t\tBalance List (" << vClient.size() << ") Client(s)\n";
	cout << "\n=========================================";
	cout << "=========================================\n";
	cout << "|" << left << setw(20) << "Account Number";
	cout << "|" << left << setw(30) << "Client Name";
	cout << "|" << left << setw(20) << "Balance";
	cout << "\n=========================================";
	cout << "=========================================\n";
}

void MidOfBalanceTable(stClient Client)
{
	cout << "|" << left << setw(20) << Client.Account_Number;
	cout << "|" << left << setw(30) << Client.Full_Name;
	cout << "|" << left << setw(20) << Client.AccountBalance;
}
```

```
void PrintMidOfBalanceTable(vector<stClient>vClient)
{
	if (vClient.size() == 0)
	{
		cout << "\t\t\tNo Clients Available In The System!";
	}
	else
	{
		for (stClient C : vClient)
		{
			MidOfBalanceTable(C);
			cout << endl;
		}
	}
}

void FooterOfBalanceTable()
{
	cout << "\n_________________________________________";
	cout << "_________________________________________\n";
}

float AllBalanceCounter(vector<stClient>vClient)
{
	stClient Client;
	float BalanceCounter = 0;

	for (stClient C : vClient)
	{
		BalanceCounter += C.AccountBalance;
	}
	return BalanceCounter;
}

void ShowBalanceScreen()
{
	vector<stClient>vClient;
	vClient = LoadClientDataFromFile(ClientsFileName);

	float BalanceCounter = AllBalanceCounter(vClient);

	HeaderOfBalanceTable(vClient);

	PrintMidOfBalanceTable(vClient);

	FooterOfBalanceTable();
	cout << "\n\t\t\t\t\t\tTotal Balance = " << BalanceCounter << endl;
}

void ShowDepositScreen()
{
	system("cls");
	cout << "\n_____________________________________________________\n";
	cout << "\t\tDeposit Screen\n";
	cout << "_____________________________________________________\n\n";
	DepositSrceen();
}
```

```
void ShowWithDrawScreen()
{
	system("cls");

	cout << "\n==========================================================\n";
	cout << "\t\tWithdraw Screen\n";
	cout << "==========================================================\n\n";

	WithdrawSrceen();
}

void ShowTotalBalancesScreen()
{
	system("cls");

	ShowBalanceScreen();
}

void GoBackToMainMenue()
{
	cout << "\n\nPress any key to go back to Main Menue...";
	system("pause>0");
	ShowMainMenueScreen();
}

void GoBackToTransactionsMenuScreen()
{
	cout << "\n\nPress any key to go back to Transactions Menue...";
	system("pause>0");
	ShowTransactionsMenuScreen();
}

void ShowEndScreen()
{
	cout << "\n==========================================================\n";
	cout << "\t\tProgram End, THANK YOU :-)" << endl;
	cout << "==========================================================\n";
}

enum enMainMenueOption {
	eListClients = 1,
	eAddNewClient = 2,
	eDeleteClient = 3,
	eUpdateClient = 4,
	eFindClient = 5,
	eTransactions = 6,
	eManageUsers = 7,
	eLogout = 8
};

enum enTransactionsMenuOption {

	eDeposit = 1,
	eWithdraw = 2,
	eTotalBalances = 3,
	eMainMenu = 4,
};
```

```
void PerformTransactionsMenuOption(enTransactionsMenuOption TransactionsMenuOption)
{
	switch (TransactionsMenuOption)
	{
	case enTransactionsMenuOption::eDeposit:
	{
		system("cls");
		ShowDepositScreen();
		GoBackToTransactionsMenuScreen();
		break;
	}
	case enTransactionsMenuOption::eWithdraw:
	{
		system("cls");
		ShowWithDrawScreen();
		GoBackToTransactionsMenuScreen();
		break;
	}
	case enTransactionsMenuOption::eTotalBalances:
	{
		system("cls");
		ShowTotalBalancesScreen();
		GoBackToTransactionsMenuScreen();
		break;
	}
	case enTransactionsMenuOption::eMainMenu:
	{
		//system("cls");
		ShowMainMenueScreen();
	}
	}
}

short ReadMainMenueOption()
{
	short Num = 0;
	cout << "Choose what do you want to do ? [1 to 8] ? ";
	cin >> Num;
	return Num;
}

short ReadTransactionsMenuOption()
{
	short Num = 0;
	cout << "Choose what do you want to do ? [1 to 4] ? ";
	cin >> Num;
	return Num;
}

void ShowTransactionsMenuScreen()
{
	if (!CheckAccessPermission(enMainMenuPermissions::pTransactions))
	{
		ShowAccessDeniedMessage();
		GoBackToMainMenue();
		return;
	}
```

```
	system("cls");
	cout << "============================================================\n";
	cout << "\t\tTransactions Menu Screen\n";
	cout << "============================================================\n";
	cout << "\t [1] Deposit.\n";
	cout << "\t [2] Withdraw.\n";
	cout << "\t [3] Total Balances.\n";
	cout << "\t [4] Main Menu.\n";
	cout << "============================================================\n";

	PerformTransactionsMenuOption((enTransactionsMenuOption)ReadTransactionsMe
nuOption());
}

//-------------------------------------------------------------------------//
//--------------- 8- Project 1 : Bank Extension 2 ---------------------//
//-------------------------------------------------------------------------//

int ReadPermissionsToSet() // مـــهـــم جـــداً
{
	int Permissions = 0;

	char Answer = 'Y';

	cout << "Do you want to give full access? (Y/N) : ";
	cin >> Answer;

	if (toupper(Answer) == 'Y')
	{
		return  -1;
	}

	cout << "\nDo you want to give access to:\n";

	cout << "Show Client List ?(Y/N): ";
	cin >> Answer;
	if (toupper(Answer) == 'Y')
	{
		Permissions += enMainMenuPermissions::pClientList;
	}

	cout << "Add New Client ?(Y/N): ";
	cin >> Answer;
	if (toupper(Answer) == 'Y')
	{
		Permissions += enMainMenuPermissions::pAddNewClien;
	}

	cout << "Delete Client ?(Y/N): ";
	cin >> Answer;
	if (toupper(Answer) == 'Y')
	{
		Permissions += enMainMenuPermissions::pDeleteClient;
	}
```

```
	cout << "Update Client ?(Y/N): ";
	cin >> Answer;
	if (toupper(Answer) == 'Y')
	{
		Permissions += enMainMenuPermissions::pUpdateClient;
	}

	cout << "Find Client ?(Y/N): ";
	cin >> Answer;
	if (toupper(Answer) == 'Y')
	{
		Permissions += enMainMenuPermissions::pFindClient;
	}

	cout << "Transactions ?(Y/N): ";
	cin >> Answer;
	if (toupper(Answer) == 'Y')
	{
		Permissions += enMainMenuPermissions::pTransactions;
	}

	cout << "Manage Users ?(Y/N): ";
	cin >> Answer;
	if (toupper(Answer) == 'Y')
	{
		Permissions += enMainMenuPermissions::pManageUsers;
	}

	return Permissions;
}

bool CheckAccessPermission(enMainMenuPermissions Permission)
{
	if (CurrentUser.Permissions == enMainMenuPermissions::eAll)
		return true;

	if ((Permission & CurrentUser.Permissions) == Permission)
		return true;

	else
		return false;
}

void ShowAccessDeniedMessage()
{
	cout << "\n-----------------------------------\n";
	cout << "Access Denied,\n";
	cout << "You don't have permission to do this,\n";
	cout << "Please contact your admin.\n";
	cout << "\n-----------------------------------\n";
}
```

```
//---------------------- Add Users To File -------------------------//

stUser CovertLineToRecord_Users(string line)
{
	vector<string>vString;
	vString = SplitString(line);

	stUser User;

	User.Username = vString[0];
	User.Password = vString[1];
	User.Permissions = stoi(vString[2]);

	return User;
}

vector<stUser> LoadUserDataFromFile(string filename)
{
	vector<stUser>vUser;

	fstream NewFile;

	NewFile.open(filename, ios::in);

	if (NewFile.is_open())
	{
		string line;
		stUser User;

		while (getline(NewFile, line))
		{
			User = CovertLineToRecord_Users(line);
			vUser.push_back(User);
		}
		NewFile.close();
	}
	return vUser;
}

bool UserExistsByUsername(string Username, string filename)
                                                    //للتحقق إذا كان الـ Username موجود أم لا
{
	vector<stUser>vUser;

	fstream NewFile;

	NewFile.open(filename, ios::in);

	if (NewFile.is_open())
	{
		string line;
		stUser User;

		while (getline(NewFile, line))
		{
			User = CovertLineToRecord_Users(line);
```

```
                if (User.Username == Username)
                {
                    NewFile.close();// مهم جداً ينقفل الملف هنا
                    return true;
                }
                vUser.push_back(User);
            }
            NewFile.close();
        }
        return false;
}

stUser ReadNewUser()
{
        stUser User;

        cout << "Enter Username: ";
        getline(cin >> ws, User.Username);

        while (UserExistsByUsername(User.Username, UsersFileName))
        {
            cout << "User with [" << User.Username << "] already exists, Enter
Another Username: ";
            getline(cin >> ws, User.Username);
        }

        cout << "Enter Password: ";
        getline(cin, User.Password);

        User.Permissions = ReadPermissionsToSet();

        return User;
}

string ConvertRecordToLine_Users(stUser User, string Delimi = "#//#")
{
        string line = "";

        line += User.Username + Delimi;
        line += User.Password + Delimi;
        line += to_string(User.Permissions);

        return line;
}

void AddUserDataToFile(string filename, string Line)
{
        fstream to_File;

        to_File.open(filename, ios::out | ios::app);

        if (to_File.is_open())
        {
            to_File << Line << endl;
            to_File.close();
        }
}
```

```
void AddNewUser()
{
	stUser User;
	User = ReadNewUser();
	AddUserDataToFile(UsersFileName, ConvertRecordToLine_Users(User));
}

void AddNewUsers()
{
	char AddMore = 'Y';
	do
	{
		cout << "\nAdding New User:\n\n";
		AddNewUser();
		cout << "\nUser Added Successfully, do you want to add more Users? (Y/N): ";
		cin >> AddMore;
	} while (toupper(AddMore) == 'Y');
}

void ShowAddNewUsersScreen()
{
	system("cls");
	cout << "\n-----------------------------------------------------\n";
	cout << "\t\tAdd New Users Screen\n";
	cout << "-----------------------------------------------------\n";
	AddNewUsers();
}
//----------------- Find User By Username ---------------------//
string EnterUserName()
{
	string Username;
	cout << "Please enter Username: ";
	cin >> Username;
	return Username;
}

void ReadUserCard(stUser User)
{
	cout << "\nThe Following are the User details:\n\n";
	cout << "Username    : " << User.Username << endl;
	cout << "Password    : " << User.Password << endl;
	cout << "Permissions : " << User.Permissions << endl;
}

bool FindUserByUserName1(stUser& User, string Username, vector<stUser>vUser)
{
	for (stUser U : vUser)
	{
		if (U.Username == Username)
		{
			User = U;
			return true;
		}
	}
	return false;
}
```

```
void ShowFindUserScreen()
{
	cout << "\n=============================================================\n";
	cout << "\t\Find User Screen\n";
	cout << "=============================================================\n\n";
	vector<stUser> vUser;
	vUser = LoadUserDataFromFile(UsersFileName);
	stUser User;
	string Username = EnterUserName();
	if (FindUserByUserName1(User, Username, vUser))
	{
		ReadUserCard(User);
	}
	else
	{
		cout << "\n\nUser with Username (" << Username << ") not found!\n";
	}
}

//---------------- Delete User By Username ------------------//

bool FindUserByUserName(string Username, vector<stUser>& vUser, stUser& User)
{
	for (stUser& U : vUser)
	{
		if (U.Username == Username)
		{
			User = U;
			return true;
		}
	}
	return false;
}

bool MarkUserForDeleteByUserName(string Username, vector<stUser>& vUser)
{
	for (stUser& U : vUser)
	{
		if (U.Username == Username)
		{
			U.MarkForDelete = true;
			return true;
		}
	}
	return false;
}

vector<stUser> SaveUserDataToFile(string filename, vector<stUser>& vUser)
{
	fstream NewFile;

	NewFile.open(filename, ios::out); // OverWrite

	string line;

	if (NewFile.is_open())
	{
```

```
            for (stUser U : vUser)
            {
                  if (U.MarkForDelete == false)
                  {
                        // We only write records that are not marked for delete.

                        line = ConvertRecordToLine_Users(U);

                        NewFile << line << endl;
                  }
            }
            NewFile.close();
      }
      return vUser;
}

bool DeleteUserByUserName(string Username, vector<stUser>& vUser)
{
      if (Username == "Admin")
      {
            cout << "\n\t==================================\n";
            cout << "\t|| You Can't Delete This User. ||";
            cout << "\n\t==================================\n";
            return false;
      }

      stUser User;

      char Answer = 'n';

      if (FindUserByUserName(Username, vUser, User))
      {
            ReadUserCard(User);

            cout << "\nAre you sure you want delete this User ? (Y / N) : ";
            cin >> Answer;

            if (toupper(Answer) == 'Y')
            {
                  MarkUserForDeleteByUserName(Username, vUser);

                  SaveUserDataToFile(UsersFileName, vUser);

                  // ReFresh Users
                  vUser = LoadUserDataFromFile(UsersFileName);

                  cout << "\n\nUser Deleted Successfully.\n";
                  return true;
            }
      }
      else
      {
            cout << "\n\nUser with Username (" << Username << ") is not
found!\n";
            return false;
      }
}
```

```
void ShowDeleteUserScreen()
{
	cout << "\n=============================================================\n";
	cout << "\t\tDelete Users Screen\n";
	cout << "=============================================================\n\n";
	vector<stUser> vUser;
	vUser = LoadUserDataFromFile(UsersFileName);
	string Username = EnterUserName();
	DeleteUserByUserName(Username, vUser);
}
//--------------- Update User By Username ------------------//
stUser ChangeUserRecord(string Username)
{
	stUser User;
	User.Username = Username;
	cout << "Enter Password ? ";
	getline(cin >> ws, User.Password);
	User.Permissions = ReadPermissionsToSet();
	return User;
}

bool UpdateUserByUserName(string Username, vector<stUser>& vUser)
{
	stUser User;
	char Answer = 'n';
	if (FindUserByUserName(Username, vUser, User))
	{
		ReadUserCard(User);

		cout << "\nAre you sure you want update this User ? (Y / N) : ";
		cin >> Answer;

		if (toupper(Answer) == 'Y')
		{
			for (stUser& U : vUser)
			{
				if (U.Username == Username)
				{
					U = ChangeUserRecord(Username);
					break;
				}
			}

			SaveUserDataToFile(UsersFileName, vUser);

			cout << "\n\nUser Updated Successfully.\n";
			return true;
		}
	}
	else
	{
		cout << "\n\nUser with Username (" << Username << ") is not
found!\n";
		return false;
	}
}
```

```
void ShowUpdateUserScreen()
{
	system("cls");
	cout << "\n=====================================================\n";
	cout << "\t\tUpdate Users Screen\n";
	cout << "=====================================================\n\n";

	vector<stUser> vUser;
	vUser = LoadUserDataFromFile(UsersFileName);

	string Username = EnterUserName();
	UpdateUserByUserName(Username, vUser);
}

//---------------------------------------------------------------------//

void GoBackToManageUsersMenuScreen()
{
	cout << "\n\nPress any key to go back to Manage Users Menue...";
	system("pause>0");
	ShowManageUsersMenuScreen();
}

void HeaderOfListUsersTable(vector<stUser>vUser)
{
	cout << "\n\t\t\t\tUsers List (" << vUser.size() << ") User(s)\n";
	cout << "\n=========================================";
	cout << "=========================================\n";
	cout << "|" << left << setw(20) << "User Name";
	cout << "|" << left << setw(30) << "Password";
	cout << "|" << left << setw(20) << "Permissions";
	cout << "\n=========================================";
	cout << "=========================================\n";
}

void MidOfListUsersTable(stUser User)
{
	cout << "|" << left << setw(20) << User.Username;
	cout << "|" << left << setw(30) << User.Password;
	cout << "|" << left << setw(20) << User.Permissions;
}

void PrintMidOfListUsersTable(vector<stUser>vUser)
{
	if (vUser.size() == 0)
	{
		cout << "\t\t\tNo Users Available In The System!";
	}
	else
	{
		for (stUser U : vUser)
		{
			MidOfListUsersTable(U);
			cout << endl;
		}
	}
}
```

```
void FooterOfListUsersTable()
{
	cout << "\n========================================";
	cout << "========================================\n";
}


float AllBalanceCounter(vector<stUser>vUser)
{
	stUser User;

	float BalanceCounter = 0;

	for (stUser U : vUser)
	{
		BalanceCounter += U.Permissions;
	}
	return BalanceCounter;
}

void ShowListUserScreen()
{
	vector<stUser>vUser;
	vUser = LoadUserDataFromFile(UsersFileName);

	//float BalanceCounter = AllBalanceCounter(vUser);

	HeaderOfListUsersTable(vUser);

	PrintMidOfListUsersTable(vUser);

	FooterOfListUsersTable();

	//cout << "\n\t\t\t\t\t\tTotal Balance = " << BalanceCounter << endl;
}

enum enManageUsersMenuOption {

	eListUsers = 1,
	eAddNewUser = 2,
	eDeleteUser = 3,
	eUpdateUser = 4,
	eFindUser = 5,
	eMainMenu1 = 6,
};

short ReadManageUsersMenuOption()
{
	short Num = 0;
	cout << "Choose what do you want to do ? [1 to 6] ? ";
	cin >> Num;
	return Num;
}
```

```
void PerformManageUsersMenuOption(enManageUsersMenuOption
ManageUsersMenuOption)
{
	switch (ManageUsersMenuOption)
	{
	case enManageUsersMenuOption::eListUsers:
	{
		system("cls");
		ShowListUserScreen();
		GoBackToManageUsersMenuScreen();
		break;
	}
	case enManageUsersMenuOption::eAddNewUser:
	{
		system("cls");
		ShowAddNewUsersScreen();
		GoBackToManageUsersMenuScreen();
		break;
	}
	case enManageUsersMenuOption::eDeleteUser:
	{
		system("cls");
		ShowDeleteUserScreen();
		GoBackToManageUsersMenuScreen();
		break;
	}
	case enManageUsersMenuOption::eUpdateUser:
	{
		system("cls");
		ShowUpdateUserScreen();
		GoBackToManageUsersMenuScreen();
		break;
	}
	case enManageUsersMenuOption::eFindUser:
	{
		system("cls");
		ShowFindUserScreen();
		GoBackToManageUsersMenuScreen();
		break;
	}
	case enManageUsersMenuOption::eMainMenu1:
	{
		//system("cls");
		ShowMainMenueScreen();
	}
	}
}

void ShowManageUsersMenuScreen()
{
	if (!CheckAccessPermission(enMainMenuPermissions::pManageUsers))
	{
		ShowAccessDeniedMessage();
		GoBackToMainMenue();
		return;
	}
```

```
	system("cls");

	cout << "============================================================\n";
	cout << "\t\tManage Users Menu Screen\n";
	cout << "============================================================\n";

	cout << "\t [1] List Users.\n";
	cout << "\t [2] Add New User.\n";
	cout << "\t [3] Delete User.\n";
	cout << "\t [4] Update User.\n";
	cout << "\t [5] Find User.\n";
	cout << "\t [6] Main Menu.\n";

	cout << "============================================================\n";

	PerformManageUsersMenuOption((enManageUsersMenuOption)ReadManageUsersMenuOption());
}
//---------------------- Login Screen ----------------------//
bool FindUserByUserNameAndPassword(string Username, string Password, stUser& User)
{
	vector<stUser>vUser;

	vUser = LoadUserDataFromFile(UsersFileName);

	for (stUser U : vUser)
	{
		if (U.Username == Username && U.Password == Password)
		{
			User = U;
			return true;
		}
	}
	return false;
}

string ReadUsername()
{
	string Username = "";
	cout << "\nPlease Enter Username: ";
	cin >> Username;
	return Username;
}

string ReadPassword()
{
	string Password = "";
	cout << "Please Enter Password: ";
	cin >> Password;
	return Password;
}

bool LoadUserInfo(string Username, string Password)
{
	if (FindUserByUserNameAndPassword(Username, Password, CurrentUser))
		return true;
	else
		return false;
}
```

```
void ShowLoginScreen()
{
	system("cls");
	cout << "\n============================\n";
	cout << "\tLogin Screen\n";
	cout << "============================\n\n";
}

void LoginSrceen()
{
	bool LoginFailed = false;
	do
	{
		ShowLoginScreen();
		if (LoginFailed)
		{
			cout << "Invalid Username/Password!\n";
		}
		string Username = ReadUsername();
		string Password = ReadPassword();
		LoginFailed = !LoadUserInfo(Username, Password);
	} while (LoginFailed);
	ShowMainMenueScreen();
}

void PerformMainMenueOption(enMainMenueOption MainMenueOption)
{
	switch (MainMenueOption)
	{
	case enMainMenueOption::eListClients:
	{
		system("cls");
		ShowAllClientsScreen();
		GoBackToMainMenue();
		break;
	}
	case enMainMenueOption::eAddNewClient:
	{
		system("cls");
		ShowAddNewClientsScreen();
		GoBackToMainMenue();
		break;
	}
	case enMainMenueOption::eDeleteClient:
	{
		system("cls");
		ShowDeleteClientScreen();
		GoBackToMainMenue();
		break;
	}
	case enMainMenueOption::eUpdateClient:
	{
		system("cls");
		ShowUpdateClientScreen();
		GoBackToMainMenue();
		break;
	}
```

```
	case enMainMenueOption::eFindClient:
	{
		system("cls");
		ShowFindClientScreen();
		GoBackToMainMenue();
		break;
	}
	case enMainMenueOption::eTransactions:
	{
		system("cls");
		ShowTransactionsMenuScreen();
		break;
	}
	case enMainMenueOption::eManageUsers:
	{
		system("cls");
		ShowManageUsersMenuScreen();
		break;
	}
	case enMainMenueOption::eLogout:
	{
		system("cls");
		LoginSrceen();
		break;
	}
	}
}

void ShowMainMenueScreen()
{
	system("cls");
	cout << "============================================================\n";
	cout << "\t\t Main Menue Screen\n";
	cout << "============================================================\n";
	cout << "\t [1] Show Client List.\n";
	cout << "\t [2] Add New Client.\n";
	cout << "\t [3] Delete Client.\n";
	cout << "\t [4] Update Client Info.\n";
	cout << "\t [5] Find Client.\n";
	cout << "\t [6] Transactions.\n";
	cout << "\t [7] Manage Users.\n";
	cout << "\t [8] Logout.\n";
	cout << "============================================================\n";
	PerformMainMenueOption((enMainMenueOption)ReadMainMenueOption());
}

int main()
{
	system("color f0");

	LoginSrceen();

	system("pause>0");
}
```

Project 2 : ATM System Requirements

```
============================
        Login Screen
============================

Enter Account Number: A150
Enter PIN: 123
```

```
=====================================================
                ATM Main Menue Screen
=====================================================
        [1] Quick Withdraw.
        [2] Normal Withdraw.
        [3] Deposit.
        [4] Check Balance.
        [5] Logout.
=====================================================
Choose what do you want to do ? [1 to 5] ?
```

```
=====================================================
                ATM Main Menue Screen
=====================================================
        [1] Quick Withdraw.
        [2] Normal Withdraw.
        [3] Deposit.
        [4] Check Balance.
        [5] Logout.
=====================================================
Choose what do you want to do ? [1 to 5] ? 1
```

```
=====================================================
                Quick Withdraw
=====================================================
        [1] 20          [2] 50
        [3] 100         [4] 200
        [5] 400         [6] 600
        [7] 800         [8] 1000
        [9] Exit
=====================================================
Your Balance is: 3850

Choose what to do from [1] to [9] ?
```

```
=====================================================
                Quick Withdraw
=====================================================
        [1] 20          [2] 50
        [3] 100         [4] 200
        [5] 400         [6] 600
        [7] 800         [8] 1000
        [9] Exit
=====================================================
Your Balance is: 3850

Choose what to do from [1] to [9] ? 2

Are you sure you want perfrom this transaction: (Y/N) ? y


Done Successfully, New balance is: 3800


Press any key to go back to Main Menue...
```

```
=====================================================
                Quick Withdraw
=====================================================
        [1] 20          [2] 50
        [3] 100         [4] 200
        [5] 400         [6] 600
        [7] 800         [8] 1000
        [9] Exit
=====================================================
Your Balance is: 700

Choose what to do from [1] to [9] ? 7
The amount exceeds your balance make another choice.
Press Any Key To Continue...
```

```
=====================================================
                ATM Main Menue Screen
=====================================================
        [1] Quick Withdraw.
        [2] Normal Withdraw.
        [3] Deposit.
        [4] Check Balance.
        [5] Logout.
=====================================================
Choose what do you want to do ? [1 to 5] ? 2
```

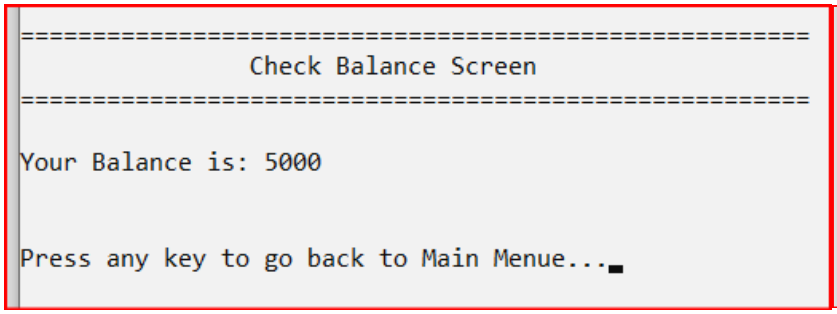
```
=====================================================
                Withdraw Screen
=====================================================


Your Balance is: 3800


Enter an amount multiple of 5's: 13

Enter an amount multiple of 5's: 4000
The amount exceeds your balance make another choice.
Press Any Key To Continue...
```

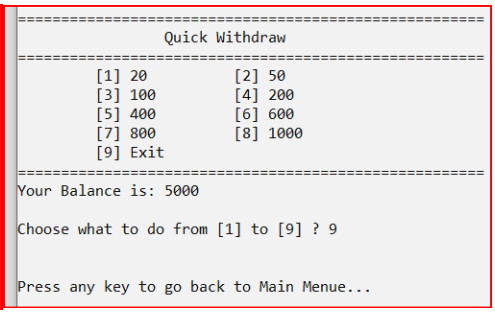
```
=====================================================
                Withdraw Screen
=====================================================


Your Balance is: 3800


Enter an amount multiple of 5's: 800

Are you sure you want perfrom this transaction: (Y/N) ? y


Done Successfully, New balance is: 3000


Press any key to go back to Main Menue...
```

```
=====================================================
                ATM Main Menue Screen
=====================================================
        [1] Quick Withdraw.
        [2] Normal Withdraw.
        [3] Deposit.
        [4] Check Balance.
        [5] Logout.
=====================================================
Choose what do you want to do ? [1 to 5] ? 3
```

```
=====================================================
                Deposit Screen
=====================================================


Your Balance is: 3000


Enter a Positive Deposit Amount: 2000

Are you sure you want perfrom this transaction: (Y/N) ? y


Done Successfully, New balance is: 5000


Press any key to go back to Main Menue...
```

```
=====================================================
                ATM Main Menue Screen
=====================================================
        [1] Quick Withdraw.
        [2] Normal Withdraw.
        [3] Deposit.
        [4] Check Balance.
        [5] Logout.
=====================================================
Choose what do you want to do ? [1 to 5] ? 4
```





Solving:

```
#include<iostream>
#include<string>
#include<fstream>
#include<vector>
#include<iomanip>
using namespace std;

//-------------------------------------------------------------------//
//--------------------- 8- Project 2 :ATM System --------------------//
//-------------------------------------------------------------------//
// نأخذ بيانات العملاء من الملف الذي أنشأناه من المشروع السابق
// عمل متغير ثابت لاسم الملف في البداية
const string ClientsFileName = "Clients.txt";

void LoginSrceen();
void ShowMainMenueScreen();
void GoBackToMainMenue();
void ShowQuickWithdrawScreen();
short ReadQuickWithdrawMenuOption();
void ShowNormalWithDrawScreen();

struct stClient {
      string Account_Number;
      string PinCode;
      string Full_Name;
      string Phone;
      float AccountBalance;

      bool MarkForDelete = false;
};

stClient CurrentClient;
stClient CovertLineToRecord_Clients(string line);

enum enMainMenueOption {
      eQuickWithdraw = 1,
      eNormalWithdraw = 2,
      eDeposit = 3,
      eCheckBalance = 4,
      eLogout = 5
};

enum enQuickWithdraw {
      e20 = 1, e50 = 2, e100 = 3, e200 = 4,
      e400 = 5, e600 = 6, e800 = 7, e1000 = 8,
      eExit = 9
};
```

```
//--------- Enter AccountNumber & Pincode --------------//

string EnterAccountNumber()
{
	string AccountNumber;
	cout << "Enter Account Number: ";
	cin >> AccountNumber;
	return AccountNumber;
}

string EnterPincode()
{
	string Pincode;
	cout << "Enter PIN: ";
	cin >> Pincode;
	return Pincode;
}

//---- Load & Convert & Split & Deposit Balance & Save & Find ----//

vector<string> SplitString(string line, string Delimi = "#//#")
{
	vector<string>vString;
	short pos = 0;
	string sWord = "";

	while ((pos = line.find(Delimi)) != std::string::npos)
	{
		sWord = line.substr(0, pos);
		if (sWord != "")
		{
			vString.push_back(sWord);
		}
		line.erase(0, pos + Delimi.length());
	}
	if (line != "")
	{
		vString.push_back(line);
	}
	return vString;
}

stClient CovertLineToRecord_Clients(string line)
{
	vector<string>vString;
	vString = SplitString(line);

	stClient Client;

	Client.Account_Number = vString[0];
	Client.PinCode = vString[1];
	Client.Full_Name = vString[2];
	Client.Phone = vString[3];
	Client.AccountBalance = stof(vString[4]);

	return Client;
}
```

```
string ConvertRecordToLine_CurrentClient(string& AccountNumber, string&
Pincode, stClient sClient, string Delimi = "#//#")
{
	string line = "";

	line += sClient.Account_Number + Delimi;
	line += sClient.PinCode + Delimi;
	line += sClient.Full_Name + Delimi;
	line += sClient.Phone + Delimi;
	if (sClient.Account_Number == AccountNumber && sClient.PinCode == Pincode)
	{
		line += to_string(CurrentClient.AccountBalance);
	}
	else
		line += to_string(sClient.AccountBalance);

	return line;
}

vector<stClient> LoadClientDataFromFile(string filename)
{
	vector<stClient>vClient;

	fstream NewFile;

	NewFile.open(filename, ios::in);

	if (NewFile.is_open())
	{
		string line;
		stClient Client;

		while (getline(NewFile, line))
		{
			Client = CovertLineToRecord_Clients(line);
			vClient.push_back(Client);
		}
		NewFile.close();
	}
	return vClient;
}

vector<stClient> SaveCruentClientDataToFile2(string& AccountNumber, string&
Pincode, string filename, vector<stClient>& vClient)
{
	fstream NewFile;

	NewFile.open(filename, ios::out); // OverWrite

	string line;

	if (NewFile.is_open())
	{
		for (stClient C : vClient)
		{
			if (C.Account_Number == AccountNumber && C.PinCode == Pincode)
			{
```

```
                    C.AccountBalance = CurrentClient.AccountBalance;
                }

                line = ConvertRecordToLine_CurrentClient(AccountNumber, Pincode, C);

                NewFile << line << endl;
            }
            NewFile.close();
        }
        return vClient;
}

bool DepositBalanceToClientByAccountNumber(string AccountNumber, float Amount,
vector<stClient>& vClient)
{
        char Answer = 'Y';

        cout << "\nAre you sure you want perfrom this transaction: (Y/N) ? ";
        cin >> Answer;

        if (toupper(Answer) == 'Y')
        {
            for (stClient& C : vClient)
            {
                if (C.Account_Number == CurrentClient.Account_Number)
                {
                    CurrentClient.AccountBalance += Amount;

                    SaveCruentClientDataToFile2(CurrentClient.Account_Number,
CurrentClient.PinCode, ClientsFileName, vClient);

                    cout << "\n\nDone Successfully, New balance is: " <<
CurrentClient.AccountBalance << endl;

                    return true;
                }
            }
            return false;
        }
}

bool FindClientByAccountNumberAndPincode(string AccountNumber, string Pincode,
stClient& Client)
{
        vector<stClient>vClient;
        vClient = LoadClientDataFromFile(ClientsFileName);

        for (stClient C : vClient)
        {
            if (C.Account_Number == AccountNumber && C.PinCode == Pincode)
            {
                Client = C;
                return true;
            }
        }
        return false;
}
```

```
//----------- Go Back To Main Menue --------------//

void GoBackToMainMenue()
{
	cout << "\n\nPress any key to go back to Main Menue...";
	system("pause>0");
	ShowMainMenueScreen();
}

//------------ Check Balance --------------------//

void ShowCheckBalanceScreen()
{
	system("cls");
	cout << "\n=========================================================\n";
	cout << "\t\tCheck Balance Screen";
	cout << "\n=========================================================\n\n";

	cout << "Your Balance is: " << CurrentClient.AccountBalance << endl;
}

//----------------- Deposit --------------------//

double ReadDepositAmount()
{
	double Amount;
	cout << "\nEnter a Positive Deposit Amount: ";
	cin >> Amount;
	while (Amount <= 0)
	{
		cout << "\nEnter a Positive Deposit Amount: ";
		cin >> Amount;
	}
	return Amount;
}

void PerformDepositOption()
{
	double DepositAmount = ReadDepositAmount();

	vector<stClient>vClient = LoadClientDataFromFile(ClientsFileName);
	DepositBalanceToClientByAccountNumber(CurrentClient.Account_Number,
DepositAmount, vClient);
}

void ShowDepositScreen()
{
	system("cls");

	cout << "\n=====================================================\n";
	cout << "\t\tDeposit Screen\n";
	cout << "=====================================================\n\n";

	cout << "\nYour Balance is: " << CurrentClient.AccountBalance << endl << endl;

	PerformDepositOption();
}
```

```
//-------------- Normal Withdraw --------------//

int ReadNormalWithDraw()
{
	int amount;

	cout << "\nEnter an amount multiple of 5's: ";
	cin >> amount;

	while (amount % 5 != 0)
	{
		cout << "\nEnter an amount multiple of 5's: ";
		cin >> amount;
	}
	return amount;
}

void PerformNormalWithDrawSrceen()
{
	int WithdrawBalance = ReadNormalWithDraw();

	if (WithdrawBalance > CurrentClient.AccountBalance)
	{
		cout << "The amount exceeds your balance make another choice.";
		cout << "\nPress Any Key To Continue...";
		system("pause>0");
		ShowNormalWithDrawScreen();
		return;
	}

	vector<stClient>vClient = LoadClientDataFromFile(ClientsFileName);

	DepositBalanceToClientByAccountNumber(CurrentClient.Account_Number,
WithdrawBalance * -1, vClient);
}

void ShowNormalWithDrawScreen()
{
	system("cls");

	cout << "\n==========================================================\n";
	cout << "\t\tWithdraw Screen\n";
	cout << "==========================================================\n\n";

	cout << "\nYour Balance is: " << CurrentClient.AccountBalance << endl <<
endl;

	PerformNormalWithDrawSrceen();
}
```

```
//------------- Quick Withdraw ---------------//
short GetQuickWithdrawAmount(short QuickWithDrawOption)
{
      switch (QuickWithDrawOption)
      {
      case 1:
            return 20;
      case 2:
            return 50;
      case 3:
            return 100;
      case 4:
            return 200;
      case 5:
            return 400;
      case 6:
            return 600;
      case 7:
            return 800;
      case 8:
            return 1000;
      default:
            return 0;
      }
}

short ReadQuickWithdrawMenuOption()
{
      short Choice = 0;
      while(Choice < 1 || Choice > 9)
      {
            cout << "\nChoose what to do from [1] to [9] ? ";
            cin >> Choice;
      }
      return Choice;
}

void PerformQuickWithDrawOption(short QuickWithDrawOption)
{
      if (QuickWithDrawOption == 9)
            return;

      short WithdrawBalance = GetQuickWithdrawAmount(QuickWithDrawOption);

      if (WithdrawBalance > CurrentClient.AccountBalance)
      {
            cout << "The amount exceeds your balance make another choice.";
            cout << "\nPress Any Key To Continue...";
            system("pause>0");
            ShowQuickWithdrawScreen();
            return;
      }

      vector<stClient>vClient = LoadClientDataFromFile(ClientsFileName);
      DepositBalanceToClientByAccountNumber(CurrentClient.Account_Number,
WithdrawBalance * -1, vClient);
}
```

```
void ShowQuickWithdrawScreen()
{
	system("cls");

	cout << "============================================================\n";
	cout << "\t\t Quick Withdraw\n";
	cout << "============================================================\n";
	cout << "\t [1] 20" << "\t\t [2] 50\n";
	cout << "\t [3] 100" << "\t [4] 200\n";
	cout << "\t [5] 400" << "\t [6] 600\n";
	cout << "\t [7] 800" << "\t [8] 1000\n";
	cout << "\t [9] Exit\n";

	cout << "============================================================\n";
	cout << "Your Balance is: " << CurrentClient.AccountBalance << endl;

	PerformQuickWithDrawOption(ReadQuickWithdrawMenuOption());
}

//------------ Main Menu --------------------//

short ReadMainMenueOption()
{
	short Num = 0;
	cout << "Choose what do you want to do ? [1 to 5] ? ";
	cin >> Num;
	return Num;
}

void PerformMainMenueOption(enMainMenueOption MainMenueOption)
{
	switch (MainMenueOption)
	{
	case enMainMenueOption::eQuickWithdraw:
	{
		system("cls");
		ShowQuickWithdrawScreen();
		GoBackToMainMenue();
		break;
	}
	case enMainMenueOption::eNormalWithdraw:
	{
		system("cls");
		ShowNormalWithDrawScreen();
		GoBackToMainMenue();
		break;
	}
	case enMainMenueOption::eDeposit:
	{
		system("cls");
		ShowDepositScreen();
		GoBackToMainMenue();
		break;
	}
```

```
	case enMainMenueOption::eCheckBalance:
	{
		system("cls");
		ShowCheckBalanceScreen();
		GoBackToMainMenue();
		break;
	}
	case enMainMenueOption::eLogout:
	{
		system("cls");
		LoginSrceen();
		break;
	}
	}
}

void ShowMainMenueScreen()
{
	system("cls");

	cout << "============================================\n";
	cout << "\t\t ATM Main Menue Screen\n";
	cout << "============================================\n";
	cout << "\t [1] Quick Withdraw.\n";
	cout << "\t [2] Normal Withdraw.\n";
	cout << "\t [3] Deposit.\n";
	cout << "\t [4] Check Balance.\n";
	cout << "\t [5] Logout.\n";
	cout << "============================================\n";

	PerformMainMenueOption((enMainMenueOption)ReadMainMenueOption());
}

//------------ Login Screen ----------------//

bool LoadClintInfo(string Accountnumber, string Pincode)
{
	if (FindClientByAccountNumberAndPincode(Accountnumber, Pincode,
CurrentClient))
		return true;
	else
		return false;
}

void ShowLoginScreen()
{
	system("cls");

	cout << "\n============================\n";
	cout << "\tLogin Screen\n";
	cout << "============================\n\n";
}
```

```
void LoginSrceen()
{
	bool LoginFailed = false;

	do
	{
		ShowLoginScreen();

		if (LoginFailed)
		{
			cout << "Invalid Account Number / Pincode!\n";
		}

		string Accountnumber = EnterAccountNumber();
		string Pincode = EnterPincode();

		LoginFailed = !LoadClintInfo(Accountnumber, Pincode);

	} while (LoginFailed);

	ShowMainMenueScreen();
}

//------------ MAIN ----------------//

int main()
{
	system("color f0");

	LoginSrceen();

	system("pause>0");
 }
```