

# الگوریتم تبدیل تاریخ شمسی به تاریخ میلادی

الگوریتم تبدیل تاریخ شمسی به تاریخ میلادی : در این مطلب دانشنامه شما علاقمندان به برنامه نویسی را با یک الگوریتم برای تبدیل تاریخ شمسی به تاریخ میلادی آشنا خواهد کرد.



الگوریتم تبدیل تاریخ شمسی به تاریخ میلادی

## الگوریتم تبدیل تاریخ شمسی به تاریخ میلادی

در این الگوریتم می خواهیم تاریخ میلادی معادل با تاریخ شمسی داده شده را بدست آوریم.

برای بدست آوردن سال میلادی معادل کافی است سال شمسی را با 621 جمع کنیم .

مهمترین قسمت نوشتن این الگوریتم تشخیص کیسه بودن سال است .

برای این کار اگر سال بدست آمده بر 100 و 400 بخشپذیر باشد ” یا بر ” 100 بخشپذیر نباشد و بر 4 بخشپذیر باشد ” آنگاه سال کیسه است ، در غیر این صورت سال کیسه نیست.

حال دو حالت پیش می آید:

سال کیسه باشد.

در این حالت اول فروردین را برابر با 20 مارس قرار می دهیم.

برای معین کردن ماه و روز ابتدا مشخص می کنیم که روز داده شده چندمین روز سال است

و 12 روز از تعداد روزها کم می کنیم، این کار را برای این انجام می دهیم که روزها را اول ماه "آوریل" محاسبه کنیم.

اگر تعداد روزها کمتر از 12 باشد آنگاه روز مورد نظر در ماه مارس قرار می گیرد و روز معادل از جمع شدن همان روز داده شده با 19 بدست می آید.

می دانیم که در سال کبیسه ماه فوریه "29" روزه می شود ، در این صورت در محاسبه ماه میلادی باید آن را لحاظ کنیم ، با توجه به این که محاسبه از ماه آوریل انجام می شود ، از ماه فوریه سال بعد برای محاسبه ماه میلادی استفاده می شود ، پس باید شرطی قرار دهیم که در صورت کبیسه شدن سال بعد ، فوریه را 29 روزه حساب کند.

سال کبیسه نباشد.

در این حالت اول فروردین را برابر با 20 مارس قرار می دهیم.

برای معین کردن ماه و روز ابتدا مشخص می کنیم که روز داده شده چندمین روز سال است و 11 روز از تعداد روزها کم می کنیم ، این کار را برای این انجام می دهیم که روزها را اول ماه "آوریل" محاسبه کنیم.

اگر تعداد روزها کمتر از 11 باشد آنگاه روز مورد نظر در ماه مارس قرار می گیرد و روز معادل از جمع شدن همان روز داده شده با 20 بدست می آید.

حال اگر تعداد روزها بیشتر از 12 یا 11 شد، تعداد روزهای ماهها را به ترتیب از "آوریل" تا "ژانویه" از کل روزها کم می کنیم .

این تفاضل را تا زمانی ادامه می دهیم تا باقیمانده از تعداد روزهای ماه بعد (که تفاضل بعدی را تشکیل می دهد) کمتر شود .

از تعداد این تفاضل ها برای بدست آوردن ماه میلادی استفاده می کنیم.

حال اگر این تعداد بیشتر از 8 شود به این معنی است که در سال بعد قرار گرفته ایم و برای بدست آوردن ماه باید 8 واحد از تعداد تفاضل ها کم کنیم و یک واحد به سال اضافه کرد و در

صورتی که کمتر از 8 شد باید 4 واحد به آن اضافه کنیم ( زیرا از ”آوریل” شروع کردیم ) تا به ماه معادل برسیم.

شبه کد الگوریتم:

0. آرایه زیر را برای نگهداری ماههای میلادی تعریف می کنیم:

{miladiMonth[12] = {30,31,30,31,31,30,31,30,31,31,28,31

;{miladiMonthLeap[12]= {30,31,30,31,31,30,31,30,31,31,29,31

1. ورودی ها— — — ShamsiYear, ShamsiMonth, ShamsiDay

miladiYear = ShamsiYear + 621.2

3. اگر ((miladiYear % 4) == 0 && (miladiYear % 100) != 0) یا

((miladiYear % 400) == 0 && (miladiYear % 100) == 0)) آنگاه

اول فروردین = 20/3 (سال کبیسه است).

marchdaydiff = 12 // اختلاف روز با ماه مارس

در غیر اینصورت

اول فروردین = 21/3 (سال کبیسه نیست).

marchdaydiff = 11 // اختلاف روز با مارس

4. اگر ShamsiMonth بین 1 تا 6 باشد آنگاه

dayCount = ((ShamsiMonth-1) \* 31) + ShamsiDay

در غیر اینصورت

dayCount = (6 \* 31) + ((ShamsiMonth - 7) \* 30) + ShamsiDay



5. اگر (dayCount < marchDayDiff) آنگاه

```
;(miladiDay = dayCount + (31 - marchDayDiff
```

```
miladiMonth = 3
```

```
miladiYear = miladiYear
```

در غیر این صورت

```
remainDay = dayCount - marchDayDiff
```

6. متغیری مانند "i" را به عنوان یک شمارنده تعریف می کنیم و مقدار اولیه آن را برابر با

"0" قرار می دهیم

7. اگر سال بعد (میلادی) کیسه نیست:

```
(([while ((remainDay > miladiMonth[i
```

```
}]
```

```
remainDay = remainDay - miladiMonth[i
```

```
++i
```

```
};
```

در غیر این صورت (اگر سال بعد کیسه است)

```
(([while ((remainDay > miladiMonthLeap[i
```

```
}]
```

```
remainDay = remainDay - miladiMonthLeap[i
```

```
++i
```

```
};
```

```
.miladiDay = remainDay;.8
```

9. اگر ( $i > 8$ ) (دقت کنید که از آوریل شروع کردیم) آنگاه

```
;miladiMonth = i - 8
```

```
;miladiYear = miladiYear + 1
```

در غیر اینصورت

```
;miladiMonth = i + 4
```

```
miladiYear = miladiYear
```

10. چاپ کن: miladiYear, miladiMonth, miladiDay

پیاده سازی الگوریتم به زبان C :

```
typedef struct _MILADIDATE  
{
```

```
    int iYear
```

```
    int iMonth
```

```
    int iDay
```

```
};MILADIDATE{
```

```
MILADIDATE ShamsiToMiladi(int ShamsiMonth, int ShamsiDay, int  
                          ShamsiYear
```

```
    }
```

```
};MILADIDATE miladiDate, farvardin1st
```

```
;int marchDayDiff, remainDay
```

```
;int dayCount, miladiYear, i
```

```
this buffer has day count of Miladi month from April to January for a none //  
    .year
```

```
;{int miladiMonth[12] = {30,31,30,31,31,30,31,30,31,31,28,31
```

```
    ;miladiYear = ShamsiYear + 621
```

```
    Detemining the Farvardin the First//
```

```
        ((if((MiladiIsLeap(miladiYear  
            }  
            }
```

```
this is a Miladi leap year so Shamsi is leap too so the 1st of Farvardin is//  
    (March 20 (3/20
```

```
        ;farvardin1st.iMonth = 3
```

```
        ;farvardin1st.iDay = 20
```

```
        ;marchDayDiff = 12
```

```
        {
```

```
        else
```

```
        }
```

```
this is not a Miladi leap year so Shamsi is not leap too so the 1st of//  
    (Farvardin is March 21 (3/21
```

```
        ;farvardin1st.iMonth = 3
```

```
        ;farvardin1st.iDay =21
```

```
        ;marchDayDiff = 11
```

```
        {
```



```

        .If next year is leap we will add one day to Feb //
                ((if(MiladiIsLeap(miladiYear+1
                                }

miladiMonth[10] = miladiMonth[10] + 1; //Adding one day to Feb
                                {

Calculate the day count for input shamsi date from 1st Farvadin//
                ((if((ShamsiMonth>=1)&&( ShamsiMonth<=6
;dayCount = ((ShamsiMonth-1) * 31) + ShamsiDay
                                else
;dayCount =(6 * 31) + ((ShamsiMonth - 7) * 30) + ShamsiDay

                Finding the correspond miladi month and day//

if (dayCount <= marchDayDiff) //So we are in 20(for leap year) or 21for
                                none leap year) to 31 march
                                }

; (miladiDate.iDay = dayCount + (31 - marchDayDiff
                                ;miladiDate.iMonth = 3
                                ;miladiDate.iYear=miladiYear
                                {
                                else
                                }

;remainDay = dayCount - marchDayDiff

```

```
        {
            i = 0; //starting from April
            ([while ((remainDay > miladiMonth[i
                }
            ];[remainDay = remainDay - miladiMonth[i
                ;++i
            {
                ;miladiDate.iDay = remainDay
            if (i > 8) // We are in the next Miladi Year
                }
                ;miladiDate.iMonth = i - 8
            ;miladiDate.iYear = miladiYear + 1
            {
                else
            }
            ;miladiDate.iMonth = i + 4
            ;miladiDate.iYear = miladiYear
            {
            {
                ;return miladiDate
```



```

    }
    .the function check a miladiyear is leap or not //
    (BOOL MiladilsLeap(int miladiYear
    }
    if(((miladiYear % 100)!= 0 && (miladiYear % 4) == 0) || ((miladiYear %
        ((100)== 0 && (miladiYear % 400) == 0
        ;return TRUE
        else
        ;return FALSE
    }
```