Top of Form



Bottom of Form

**Thuật toán tính âm lịch**

**Hồ Ngọc Đức**

Bài viết sau giới thiệu cách tính âm lịch Việt Nam và mô tả một số thuật toán dùng để chuyển đổi giữa ngày dương lịch và ngày âm lịch. Các thuật toán mô tả ở đây đã được đơn giản hóa nhiều để bạn đọc tiện theo dõi và dễ dàng sử dụng vào việc lập trình, do đó độ chính xác của chúng thấp hơn độ chính xác của chương trình âm lịch trực tuyến tại [**http://www.informatik.uni-leipzig.de/~duc/amlich/**](http://www.informatik.uni-leipzig.de/~duc/amlich/). (Một phiên bản cũ của bài viết này giới thiệu vài thuật toán hơi khác, có thể khó thực hiện hơn một chút. Bản cũ này có thể xem [tại đây](https://www.informatik.uni-leipzig.de/~duc/amlich/calrules_v02.html).)

[If you cannot read Vietnamese: [Old version in English](https://www.informatik.uni-leipzig.de/~duc/amlich/calrules_en.html)]

**Quy luật của âm lịch Việt Nam**

Âm lịch Việt Nam là một loại lịch thiên văn. Nó được tính toán dựa trên sự chuyển động của mặt trời, trái đất và mặt trăng. Ngày tháng âm lịch được tính dựa theo các nguyên tắc sau:

1. Ngày đầu tiên của tháng âm lịch là ngày chứa điểm Sóc
2. Một năm bình thường có 12 tháng âm lịch, một năm nhuận có 13 tháng âm lịch
3. Đông chí luôn rơi vào tháng 11 âm lịch
4. Trong một năm nhuận, nếu có 1 tháng không có Trung khí thì tháng đó là tháng nhuận. Nếu nhiều tháng trong năm nhuận đều không có Trung khí thì chỉ tháng đầu tiên sau Đông chí là tháng nhuận
5. Việc tính toán dựa trên kinh tuyến 105° đông.

Sóc là thời điểm hội diện, đó là khi trái đất, mặt trăng và mặt trời nằm trên một đường thẳng và mặt trăng nằm giữa trái đất và mặt trời. (Như thế góc giữa mặt trăng và mặt trời bằng 0 độ). Gọi là "hội diện" vì mặt trăng và mặt trời ở cùng một hướng đối với trái đất. Chu kỳ của điểm Sóc là khoảng 29,5 ngày. Ngày chứa điểm Sóc được gọi là ngày Sóc, và đó là ngày bắt đầu tháng âm lịch.

Trung khí là các điểm chia đường hoàng đạo thành 12 phần bằng nhau. Trong đó, bốn Trung khí giữa bốn mùa là đặc biệt nhất: Xuân phân (khoảng 20/3), Hạ chí (khoảng 22/6), Thu phân (khoảng 23/9) và Đông chí (khoảng 22/12).

Bởi vì dựa trên cả mặt trời và mặt trăng nên lịch Việt Nam không phải là thuần âm lịch mà là âm-dương-lịch. Theo các nguyên tắc trên, để tính ngày tháng âm lịch cho một năm bất kỳ trước hết chúng ta cần xác định những ngày nào trong năm chứa các thời điểm Sóc (New moon) . Một khi bạn đã tính được ngày Sóc, bạn đã biết được ngày bắt đầu và kết thúc của một tháng âm lịch: ngày mùng một của tháng âm lịch là ngày chứa điểm sóc. Sau khi đã biết ngày bắt đầu/kết thúc các tháng âm lịch, ta tính xem các Trung khí (Major solar term) rơi vào tháng nào để từ đó xác định tên các tháng và tìm tháng nhuận.

Đông chí luôn rơi vào tháng 11 của năm âm lịch. Bởi vậy chúng ta cần tính 2 điểm sóc: Sóc A ngay trước ngày Đông chí thứ nhất và Sóc B ngay trước ngày Đông chí thứ hai. Nếu khoảng cách giữa A và B là dưới 365 ngày thì năm âm lịch có 12 tháng, và những tháng đó có tên là: tháng 11, tháng 12, tháng 1, tháng 2, …, tháng 10. Ngược lại, nếu khoảng cách giữa hai sóc A và B là trên 365 ngày thì năm âm lịch này là năm nhuận, và chúng ta cần tìm xem đâu là tháng nhuận. Để làm việc này ta xem xét tất cả các tháng giữa A và B, tháng đầu tiên không chứa Trung khí sau ngày Đông chí thứ nhất là tháng nhuận. Tháng đó sẽ được mang tên của tháng trước nó kèm chữ "nhuận".

Khi tính ngày Sóc và ngày chứa Trung khí bạn cần lưu ý xem xét chính xác múi giờ. Đây là lý do tại sao có một vài điểm khác nhau giữa lịch Việt Nam và lịch Trung Quốc.Ví dụ, nếu bạn biết thời điểm hội diện là vào lúc yyyy-02-18 16:24:45 GMT thì ngày Sóc của lịch Việt Nam là 18 tháng 2, bởi vì 16:24:45 GMT là 23:24:45 cùng ngày, giờ Hà nội (GMT+7, kinh tuyến 105° đông). Tuy nhiên theo giờ Bắc Kinh (GMT+8, kinh tuyến 120° đông) thì Sóc là lúc 00:24:45 ngày yyyy-02-19, do đó tháng âm lịch của Trung Quốc lại bắt đầu ngày yyyy-02-19, chậm hơn lịch Việt Nam 1 ngày.

**Ví dụ 1: Âm lịch năm 1984**

Chúng ta áp dụng quy luật trên để tính âm lịch Việt nam năm 1984.

* Sóc A (ngay trước Đông chí năm 1983) rơi vào ngày 4/12/1983, Sóc B (ngay trước Đông chí năm 1984) vào ngày 23/11/1984.
* Giữa A và B là khoảng 355 ngày, như thế năm âm lịch 1984 là năm thường. Tháng 11 âm lịch của năm trước kéo dài từ 4/12/1983 đến 2/01/1984, tháng 12 âm từ 3/1/1984 đến 1/2/1984, tháng Giêng từ 2/2/1984 đến 1/3/1984 v.v.

**Ví dụ 2: Âm lịch năm 2004**

* Sóc A - điểm sóc cuối cùng trước Đông chí 2003 - rơi vào ngày 23/11/2003. Sóc B (ngay trước Đông chí năm 2004) rơi vào ngày 12/12/2004.
* Giữa 2 ngày này là khoảng 385 ngày, như vậy năm âm lịch 2004 là năm nhuận. Tháng 11 âm của năm 2003 bắt đầu vào ngày chứa Sóc A, tức ngày 23/11/2003.
* Tháng âm lịch đầu tiên sau đó mà không chứa Trung khí là tháng từ 21/3/2004 đến 18/4/2004 (Xuân phân rơi vào 20/3/2004, còn Cốc vũ là 19/4/2004). Như thế tháng ấy là tháng nhuận.
* Từ 23/11/2003 đến 21/3/2004 là khoảng 120 ngày, tức 4 tháng âm lịch: tháng 11, 12, 1 và 2. Như vậy năm 2004 có tháng 2 nhuận.

**Thuật toán chuyển đổi giữa ngày dương và âm**

Trong tính toán thiên văn người ta lấy ngày 1/1/4713 trước công nguyên của lịch Julius (tức ngày 24/11/4714 trước CN theo lịch Gregorius) làm điểm gốc. Số ngày tính từ điểm gốc này gọi là số ngày Julius (Julian day number) của một thời điểm. Ví dụ, số ngày Julius của 1/1/2000 là 24515455.

Dùng các công thức sau ta có thể chuyển đổi giữa ngày/tháng/năm và số ngày Julius. Phép chia ở 2 công thức sau được hiểu là chia số nguyên, bỏ phần dư: 23/4=5.

**Đổi ngày dd/mm/yyyy ra số ngày Julius jd**

a = (14 - mm) / 12

y = yy+4800-a

m = mm+12\*a-3

Lịch Gregory:

jd = dd + (153\*m+2)/5 + 365\*y + y/4 - y/100 + y/400 - 32045

Lịch Julius:

jd = dd + (153\*m+2)/5 + 365\*y + y/4 - 32083

**Đổi số ngày Julius jd ra ngày dd/mm/yyyy**

Lịch Gregory (jd lớn hơn 2299160):

a = jd + 32044;

b = (4\*a+3)/146097;

c = a - (b\*146097)/4;

Lịch Julius:

b = 0;

c = jd + 32082;

Công thức cho cả 2 loại lịch:

d = (4\*c+3)/1461;

e = c - (1461\*d)/4;

m = (5\*e+2)/153;

dd = e - (153\*m+2)/5 + 1;

mm = m + 3 - 12\*(m/10);

yy = b\*100 + d - 4800 + m/10;

Nếu ngôn ngữ lập trình bạn dùng không hỗ trợ phép chia số nguyên bỏ phần dư (VD: JavaScript), bạn có thể định nghĩa một hàm INT(x) để lấy số nguyên lớn nhất không vượt quá x: INT(5)=5, INT(3.2)=3, INT(-5)=-5, INT(-3.2)=-4 v.v. Khi đó, INT(m/10) sẽ trả lại kết quả của phép chia số nguyên. (Nhiều ngôn ngữ có sẵn hàm floor() cho phép làm việc này.)

Các phép chuyển đổi giữa ngày tháng và số ngày Julius có thể được thực hiện với mã JavaScript như sau:

**function jdFromDate(dd, mm, yy)**

var a, y, m, jd;

a = INT((14 - mm) / 12);

y = yy+4800-a;

m = mm+12\*a-3;

jd = dd + INT((153\*m+2)/5) + 365\*y + INT(y/4) - INT(y/100) + INT(y/400) - 32045;

if (jd < 2299161) {

jd = dd + INT((153\*m+2)/5) + 365\*y + INT(y/4) - 32083;

}

return jd;

**function jdToDate(jd)**

var a, b, c, d, e, m, day, month, year;

if (jd > 2299160) { // After 5/10/1582, Gregorian calendar

a = jd + 32044;

b = INT((4\*a+3)/146097);

c = a - INT((b\*146097)/4);

} else {

b = 0;

c = jd + 32082;

}

d = INT((4\*c+3)/1461);

e = c - INT((1461\*d)/4);

m = INT((5\*e+2)/153);

day = e - INT((153\*m+2)/5) + 1;

month = m + 3 - 12\*INT(m/10);

year = b\*100 + d - 4800 + INT(m/10);

return new Array(day, month, year);

Trong các công thức sau, timeZone là số giờ chênh lệch giữa giờ địa phương và giờ UTC (hay GMT). (Để tính lịch Việt Nam, lấy timeZone = 7.0). Các phương pháp sau được giới thiệu với mã JavaScript. Bạn có thể tải [thư viện JavaScript](https://www.informatik.uni-leipzig.de/~duc/amlich/amlich-aa98.js) hoặc [thư viện PHP](https://www.informatik.uni-leipzig.de/~duc/amlich/amlich-php.txt) hoàn chỉnh để tham khảo.

**Tính ngày Sóc**

Như trên đã nói, để tính được âm lịch trước hết ta cần xác định các tháng âm lịch bắt đầu vào ngày nào.

Thuật toán sau tính ngày Sóc thứ k kể từ điểm Sóc ngày 1/1/1900. Kết quả trả về là số ngày Julius của ngày Sóc cần tìm.

**function getNewMoonDay(k, timeZone)**

var T, T2, T3, dr, Jd1, M, Mpr, F, C1, deltat, JdNew;

T = k/1236.85; // Time in Julian centuries from 1900 January 0.5

T2 = T \* T;

T3 = T2 \* T;

dr = PI/180;

Jd1 = 2415020.75933 + 29.53058868\*k + 0.0001178\*T2 - 0.000000155\*T3;

Jd1 = Jd1 + 0.00033\*Math.sin((166.56 + 132.87\*T - 0.009173\*T2)\*dr); // Mean new moon

M = 359.2242 + 29.10535608\*k - 0.0000333\*T2 - 0.00000347\*T3; // Sun's mean anomaly

Mpr = 306.0253 + 385.81691806\*k + 0.0107306\*T2 + 0.00001236\*T3; // Moon's mean anomaly

F = 21.2964 + 390.67050646\*k - 0.0016528\*T2 - 0.00000239\*T3; // Moon's argument of latitude

C1=(0.1734 - 0.000393\*T)\*Math.sin(M\*dr) + 0.0021\*Math.sin(2\*dr\*M);

C1 = C1 - 0.4068\*Math.sin(Mpr\*dr) + 0.0161\*Math.sin(dr\*2\*Mpr);

C1 = C1 - 0.0004\*Math.sin(dr\*3\*Mpr);

C1 = C1 + 0.0104\*Math.sin(dr\*2\*F) - 0.0051\*Math.sin(dr\*(M+Mpr));

C1 = C1 - 0.0074\*Math.sin(dr\*(M-Mpr)) + 0.0004\*Math.sin(dr\*(2\*F+M));

C1 = C1 - 0.0004\*Math.sin(dr\*(2\*F-M)) - 0.0006\*Math.sin(dr\*(2\*F+Mpr));

C1 = C1 + 0.0010\*Math.sin(dr\*(2\*F-Mpr)) + 0.0005\*Math.sin(dr\*(2\*Mpr+M));

if (T < -11) {

deltat= 0.001 + 0.000839\*T + 0.0002261\*T2 - 0.00000845\*T3 - 0.000000081\*T\*T3;

} else {

deltat= -0.000278 + 0.000265\*T + 0.000262\*T2;

};

JdNew = Jd1 + C1 - deltat;

return INT(JdNew + 0.5 + timeZone/24)

Với hàm này ta có thể tính được tháng âm lịch chứa ngày N bắt đầu vào ngày nào: giữa ngày 1/1/1900 (số ngày Julius: 2415021) và ngày N có khoảng k=INT((N-2415021)/29.530588853) tháng âm lịch, như thế dùng hàm **getNewMoonDay** sẽ biết ngày đầu tháng âm lịch chứa ngày N, từ đó ta biết ngày N là mùng mấy âm lịch.

**Tính tọa độ mặt trời**

Để biết Trung khí nào nằm trong tháng âm lịch nào, ta chỉ cần tính xem mặt trời nằm ở khoảng nào trên đường hoàng đạo vào thời điểm bắt đầu một tháng âm lịch. Ta chia đường hoàng đạo làm 12 phần và đánh số các cung này từ 0 đến 11: từ Xuân phân đến Cốc vũ là 0; từ Cốc vũ đến Tiểu mãn là 1; từ Tiểu mãn đến Hạ chí là 2; v.v.. Cho jdn là số ngày Julius của bất kỳ một ngày, phương pháp sau này sẽ trả lại số cung nói trên.

**function getSunLongitude(jdn, timeZone)**

var T, T2, dr, M, L0, DL, L;

T = (jdn - 2451545.5 - timeZone/24) / 36525; // Time in Julian centuries from 2000-01-01 12:00:00 GMT

T2 = T\*T;

dr = PI/180; // degree to radian

M = 357.52910 + 35999.05030\*T - 0.0001559\*T2 - 0.00000048\*T\*T2; // mean anomaly, degree

L0 = 280.46645 + 36000.76983\*T + 0.0003032\*T2; // mean longitude, degree

DL = (1.914600 - 0.004817\*T - 0.000014\*T2)\*Math.sin(dr\*M);

DL = DL + (0.019993 - 0.000101\*T)\*Math.sin(dr\*2\*M) + 0.000290\*Math.sin(dr\*3\*M);

L = L0 + DL; // true longitude, degree

L = L\*dr;

L = L - PI\*2\*(INT(L/(PI\*2))); // Normalize to (0, 2\*PI)

return INT(L / PI \* 6)

Với hàm này ta biết được một tháng âm lịch chứa Trung khí nào. Giả sử một tháng âm lịch bắt đầu vào ngày N1 và tháng sau đó bắt đầu vào ngày N2 và hàm getSunLongitude cho kết quả là 8 với N1 và 9 với N2. Như vậy tháng âm lịch bắt đầu ngày N1 là tháng chứa Đông chí: trong khoảng từ N1 đến N2 có một ngày mặt trời di chuyển từ cung 8 (sau Tiểu tuyết) sang cung 9 (sau Đông chí). Nếu hàm getSunLongitude trả lại cùng một kết quả cho cả ngày bắt đầu một tháng âm lịch và ngày bắt đầu tháng sau đó thì tháng đó không có Trung khí và như vậy có thể là tháng nhuận.

**Tìm ngày bắt đầu tháng 11 âm lịch**

Đông chí thường nằm vào khoảng 19/12-22/12, như vậy trước hết ta tìm ngày Sóc trước ngày 31/12. Nếu tháng bắt đầu vào ngày đó không chứa Đông chí thì ta phải lùi lại 1 tháng nữa.

**function getLunarMonth11(yy, timeZone)**

var k, off, nm, sunLong;

off = jdFromDate(31, 12, yy) - 2415021;

k = INT(off / 29.530588853);

nm = getNewMoonDay(k, timeZone);

sunLong = getSunLongitude(nm, timeZone); // sun longitude at local midnight

if (sunLong >= 9) {

nm = getNewMoonDay(k-1, timeZone);

}

return nm;

**Xác định tháng nhuận**

Nếu giữa hai tháng 11 âm lịch (tức tháng có chứa Đông chí) có 13 tháng âm lịch thì năm âm lịch đó có tháng nhuận. Để xác định tháng nhuận, ta sử dụng hàm **getSunLongitude** như đã nói ở trên. Cho a11 là ngày bắt đầu tháng 11 âm lịch mà một trong 13 tháng sau đó là tháng nhuận. Hàm sau cho biết tháng nhuận nằm ở vị trí nào sau tháng 11 này.

**function getLeapMonthOffset(a11, timeZone)**

var k, last, arc, i;

k = INT((a11 - 2415021.076998695) / 29.530588853 + 0.5);

last = 0;

i = 1; // We start with the month following lunar month 11

arc = getSunLongitude(getNewMoonDay(k+i, timeZone), timeZone);

do {

last = arc;

i++;

arc = getSunLongitude(getNewMoonDay(k+i, timeZone), timeZone);

} while (arc != last && i < 14);

return i-1;

Giả sử hàm **getLeapMonthOffset** trả lại giá trị 4, như thế tháng nhuận sẽ là tháng sau tháng 2 thường. (Tháng thứ 4 sau tháng 11 đáng ra là tháng 3, nhưng vì đó là tháng nhuận nên sẽ lấy tên của tháng trước đó tức tháng 2, và tháng thứ 5 sau tháng 11 mới là tháng 3).

**Đổi ngày dương dd/mm/yyyy ra ngày âm**

Với các phương pháp hỗ trợ trên ta có thể đổi ngày dương dd/mm/yy ra ngày âm dễ dàng. Trước hết ta xem ngày monthStart bắt đầu tháng âm lịch chứa ngày này là ngày nào (dùng hàm getNewMoonDay như trên đã nói). Sau đó, ta tìm các ngày a11 và b11 là ngày bắt đầu các tháng 11 âm lịch trước và sau ngày đang xem xét. Nếu hai ngày này cách nhau dưới 365 ngày thì ta chỉ còn cần xem monthStart và a11 cách nhau bao nhiêu tháng là có thể tính được dd/mm/yy nằm trong tháng mấy âm lịch. Ngược lại, nếu a11 và b11 cách nhau khoảng 13 tháng âm lịch thì ta phải tìm xem tháng nào là tháng nhuận và từ đó suy ra ngày đang tìm nằm trong tháng nào.

**function convertSolar2Lunar(dd, mm, yy, timeZone)**

var k, dayNumber, monthStart, a11, b11, lunarDay, lunarMonth, lunarYear, lunarLeap;

dayNumber = jdFromDate(dd, mm, yy);

k = INT((dayNumber - 2415021.076998695) / 29.530588853);

monthStart = getNewMoonDay(k+1, timeZone);

if (monthStart > dayNumber) {

monthStart = getNewMoonDay(k, timeZone);

}

a11 = getLunarMonth11(yy, timeZone);

b11 = a11;

if (a11 >= monthStart) {

lunarYear = yy;

a11 = getLunarMonth11(yy-1, timeZone);

} else {

lunarYear = yy+1;

b11 = getLunarMonth11(yy+1, timeZone);

}

lunarDay = dayNumber-monthStart+1;

diff = INT((monthStart - a11)/29);

lunarLeap = 0;

lunarMonth = diff+11;

if (b11 - a11 > 365) {

leapMonthDiff = getLeapMonthOffset(a11, timeZone);

if (diff >= leapMonthDiff) {

lunarMonth = diff + 10;

if (diff == leapMonthDiff) {

lunarLeap = 1;

}

}

}

if (lunarMonth > 12) {

lunarMonth = lunarMonth - 12;

}

if (lunarMonth >= 11 && diff < 4) {

lunarYear -= 1;

}

**Đổi âm lịch ra dương lịch**

Cách làm cũng tương tự như đổi ngày dương sang ngày âm.

**function convertLunar2Solar(lunarDay, lunarMonth, lunarYear, lunarLeap, timeZone)**

var k, a11, b11, off, leapOff, leapMonth, monthStart;

if (lunarMonth < 11) {

a11 = getLunarMonth11(lunarYear-1, timeZone);

b11 = getLunarMonth11(lunarYear, timeZone);

} else {

a11 = getLunarMonth11(lunarYear, timeZone);

b11 = getLunarMonth11(lunarYear+1, timeZone);

}

off = lunarMonth - 11;

if (off < 0) {

off += 12;

}

if (b11 - a11 > 365) {

leapOff = getLeapMonthOffset(a11, timeZone);

leapMonth = leapOff - 2;

if (leapMonth < 0) {

leapMonth += 12;

}

if (lunarLeap != 0 && lunarMonth != leapMonth) {

return new Array(0, 0, 0);

} else if (lunarLeap != 0 || off >= leapOff) {

off += 1;

}

}

k = INT(0.5 + (a11 - 2415021.076998695) / 29.530588853);

monthStart = getNewMoonDay(k+off, timeZone);

return jdToDate(monthStart+lunarDay-1);

**Tính ngày thứ và Can-Chi cho ngày và tháng âm lịch**

Ngày thứ lặp lại theo chu kỳ 7 ngày, như thế để biết một ngày d/m/y bất kỳ là thứ mấy ta chỉ việc tìm số dư của số ngày Julius của ngày này cho 7.

Để tính Can của năm Y, tìm số dư của Y+6 chia cho 10. Số dư 0 là Giáp, 1 là Ất v.v. Để tính Chi của năm, chia Y+8 cho 12. Số dư 0 là Tý, 1 là Sửu, 2 là Dần v.v.

Hiệu Can-Chi của ngày lặp lại theo chu kỳ 60 ngày, như thế nó cũng có thể tính được một cách đơn giản. Cho N là số ngày Julius của ngày dd/mm/yyyy. Ta chia N+9 cho 10. Số dư 0 là Giáp, 1 là Ất v.v. Để tìm Chi, chia N+1 cho 12; số dư 0 là Tý, 1 là Sửu v.v.

Trong một năm âm lịch, tháng 11 là tháng Tý, tháng 12 là Sửu, tháng Giêng là tháng Dần v.v. Can của tháng M năm Y âm lịch được tính theo công thức sau: chia Y\*12+M+3 cho 10. Số dư 0 là Giáp, 1 là Ất v.v.

Ví dụ, Can-Chi của tháng 3 âm lịch năm Giáp Thân 2004 là Mậu Thìn: tháng 3 âm lịch là tháng Thìn, và (2004\*12+3+3) % 10 = 24054 % 10 = 4, như vậy Can của tháng là Mậu.

Một tháng nhuận không có tên riêng mà lấy tên của tháng trước đó kèm thêm chữ "Nhuận", VD: tháng 2 nhuận năm Giáp Thân 2004 là tháng Đinh Mão nhuận.

**Tài liệu tham khảo**

* Edward M. Reingold and Nachum Dershowitz, [Calendrical Calculations](http://emr.cs.iit.edu/~reingold/calendars.shtml)
* Helmer Aslaksen, [The Mathematics of the Chinese Calendar](http://www.math.nus.edu.sg/aslaksen/calendar/)
* [Eric Weisstein's World of Astronomy](http://scienceworld.wolfram.com/astronomy/JulianDate.html)
* [Calendar FAQ](http://www.tondering.dk/claus/calendar.html)
* [projectpluto.com](http://www.projectpluto.com/) (Open source implementation of many astronomical functions)
* Hồ Ngọc Đức, [Âm lịch Việt Nam qua các thời kỳ lịch sử](https://www.informatik.uni-leipzig.de/~duc/amlich/histcal.html)
* Hồ Ngọc Đức, [VietCalendar.java](https://www.informatik.uni-leipzig.de/~duc/amlich/VietCalendar.java) - Một chương trình Java nhỏ (300 dòng) để chuyển đổi âm dương lịch
* Hồ Ngọc Đức, [JavaScript lunar calendar](https://www.informatik.uni-leipzig.de/~duc/amlich/JavaScript/) - Chương trình âm lịch Việt Nam bằng JavaScript (sử dụng bảng tính sẵn cho thời kỳ 1800-2199)
* Tính âm lịch dùng các ngôn ngữ khác: [JavaScript](https://www.informatik.uni-leipzig.de/~duc/amlich/amlich-aa98.js), [PHP](https://www.informatik.uni-leipzig.de/~duc/amlich/amlich-php.txt), [Visual Basic](https://www.informatik.uni-leipzig.de/~duc/amlich/LunarSolar.bas), [Python](https://www.informatik.uni-leipzig.de/~duc/amlich/AL.py) (chương trình Visual Basic và Python do Nguyễn Quốc Sản đóng góp)

Sửa đổi lần cuối: 18/08/2008

Top of Form

Bottom of Form