BÁO CÁO ĐỒ ÁN 2

MÔN: KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ

THÀNH VIÊN NHÓM:

1612700 Nguyễn Ngô Tín

1612714 Nguyễn Hồng Tới

1612726 Nguyễn Minh Trí

Mục Lục:

I. Thông tin thành viên và bảng phân công công việc:

MSSV	Họ và tên	Công việc	Đóng góp	Hoàn thành	Ký tên
1612700	Nguyễn Ngô Tín	- Viết hàm: • **char* Convert(char *Time,char type)** • **bool LeapYear(int Year)** • **int KiemTraSo(int ngay, int thang, int nam)**	33%	100%	
1612714	Nguyễn Hồng Tới	- Viết báo cáo - Viết hàm: • **char* Weekday(char* Time)** • **int NextLeapYear(char *Time)**	33%	100%	
1612726	Nguyễn Minh Trí	- Viết main - Viết hàm: • **char* Date(int Day, int Month, int Year, char* a3)** • **int KhoangCach(char* Time1, char* Time2)** • **int KiemTraSo(int ngay, int thang, int nam)**	33%	100%	

II. Nội dung đồ án:

1. **Môi trường lập trình:** MARS 4.0.1
2. **Qui tắc khi viết và gọi hàm trong MIPS:**

- MIPS hỗ trợ một số thanh ghi để lưu trữ các dữ liệu phục vụ cho thủ tục: đối số($a0, $a1, $a2), kết quả trả về($v0, $v1), địa chỉ trả về($ra).
- Trước khi gọi hàm ta cần đặt các giá trị vào thanh ghi $a0-$a3
- Gọi hàm bằng lệnh jal sẽ làm tiết kiệm địa chỉ trả về trong $ra, địa chỉ trả về là PC + 4, lệnh PC là địa chỉ của lệnh jal.
- Trả về kết quả trong thanh ghi $v0 - $v1 bằng lệnh jr $ra, chỉ có một lệnh jr $ra trong mỗi thủ tục.
- Nếu gọi thủ tục con để hỗ trợ thủ tục cha thì:
  - Lưu địa chỉ trả về của thủ tục cha ($ra) vào ngăn xếp.
  - Trong thủ tục con, ta khởi tạo ngăn xếp và lưu trong ngắn xếp các thanh ghi có thể bị thay đổi của thủ tục cha.
  - Sau đó khôi phục dữ liệu đã lưu trong ngăn xếp và phục hồi ngăn xếp trong thủ tục con.
  - Trả về giá trị của thủ tục con bằng lệnh jr $ra.

3. **Các hàm đã cài đặt:**

a) Char* Date(int Day, int Month, int Year, char* a3)

- Mô tả cách thức cài đặt:

Hàm được cài đặt với 4 tham số đầu vào:

- $a0 ứng với ngày
- $a1 ứng với tháng
- $a2 ứng với năm
- $a3 giữ chuỗi Time để lưu
- Các tham số đầu vào đã được kiểm tra đúng
- Trước tiên ta lấy $a0(ngày) chia cho 10 và lấy thương cho vào 0($a3) và lấy dư cho vào 1($a3), và thêm dấu "/" vào 2($a3)
- Tiếp theo ta lấy $a1(tháng) chia cho 10 và lấy thương cho tiếp vào 3($a3) và lấy dư cho vào 4($a3), và thêm dấu "/" vào 5($a3)
- Với $a2(năm) thì ta chia cho 1000 lấy thương đưa vào 6($a3) rồi lấy phần dư chia 100 lấy thương đưa vào 7($a3) và lấy phần dư chia 10 lấy thương đưa vào 8($a3) và lấy phần dư đưa vào 9($a3)
- Sau khi thực hiện ta được chuỗi Time: DD/MM/YYYY

b) Char * Convert(char *Time,char type)

- Mô tả cách thức cài đặt:
    - Từ chuỗi char* Time đầu vào có định dạng là DD/MM/YYYY ta sẽ loadbyte từng kí tự của chuôi Time đầu vào 1 thanh ghi, sau đó cất lần lượt các kí tự này vào stack, ta chỉ cất các kí tự nằm trong khoảng 48 đến 58, các kí tự / sẽ được bỏ qua. Sau khi cất lần lượt vào stack thì ta sẽ có 1 stack như sau DDMMYYYY($sp) $sp sẽ ở ngay sau kí tự Y cuối cùng. Nhờ vào việc lưu lại ở các thanh ghi $s0, $s1, $s2 lần lượt là số kí tự của ngày tháng năm thì ta sẽ dễ dàng trong việc lấy ra các kí tự để sắp xếp vào chuỗi Time kết quả. Ứng với từng type ta sẽ có cách lấy ra khác nhau, ví dụ như muốn chuyển thành kiểu MM/DD/YYYY thì đầu tiên ta phải cho thanh ghi $sp về ngay sau kí tự Month đầu tiên,bằng cách lấy $sp - 1 + số kí tự của năm + số kí tự của tháng, như trường hợp trên thì ta sẽ có stack như sau DDM($sp)MYYYY khi đó ta sẽ đọc tháng ra, rồi thực hiện phép cộng trừ trên $sp để có stack là D($sp)DMMYYYY để đọc ngày ra,rồi lại tiếp tục DDMMY($sp)YYY để đọc năm, các kí tự sau khi đọc sẽ được cho vào Time và cứ như thể ta sẽ được chuỗi Time mới.

c) char* Weekday(char* Time):

- Mô tả cách thức cài đặt:
    - Hàm được cài đặt với các tham số ngày, tháng, năm lấy từ chuỗi Time.
    - Dựa vào công thức:

        $$w = d + m + y + y/4 + c \bmod 7$$

      Trong đó:

        d: ngày
        m: tháng tương ứng trong bảng Month
        y: là 2 số cuối của năm
        c: là thế kỷ

      Kết quả: w thuộc tập giá trị {0,1,2,3,4, 5, 6} tương ứng các ngày trong tuần {Sun, Mon, Tues, Wed, Thurs, Fri, Sat}

      Bảng Month:

Month	m	Leap years
January	0	6
February	3	2
March	3	
April	6	
May	1	
June	4	
July	6	
August	2	
September	5	
October	0	
November	3	
December	5	

- Thế kỷ (c) bằng thương của $a2(năm) chia hết cho 100, ngược lại nếu không chia hết thì thế kỷ (c) bằng thương của $a2(năm) chia 100 cộng 1.
- Tính y bằng phần dư của $a2(năm) chia cho 100, khi đó ta tính được y/4 bằng cách lấy thương của y chia 4.
- Khi có c, y ta tính được w và lấy w mod 7. Phần dư của w mod 7 sẽ thuộc tập giá trị {0,1,2,3,4, 5, 6} từ đó ta có được thứ trong tuần.

d) bool LeapYear(int Year)

- Mô tả cách thức cài đặt:
  - Hàm được cài đặt với input là $a0(Year)
  - Trước tiên ta lấy $a0(Year) chia cho 400 và kiểm tra xem có chia hết hay không, nếu chia hết thì trả về 1(năm nhuận)
  - Nếu không chia hết ta sẽ kiểm tra tiếp $a0(Year) có chia hết cho 100 hay không, nếu chia hết thì trả về 0(năm không nhuận)
  - Nếu không chia hết ta kiểm tra tiếp xem nó có chia hết cho 4 hay không, nếu chia hết trả về 1(năm nhuận) và trả về 0(năm không nhuận) nếu không chia hết.

e) int KhoangCach(char* Time1, char* Time2):

- Mô tả cách thức cài đặt:

- Trước tiên ta sẽ lấy Year1 và Year2 của 2 chuỗi Time, sau đó ta so sánh 2 chuỗi đó với nhau, nếu Year bằng nhau ta trả về KC = 0
- Nếu không bằng sau ta kiểm tra xem năm nào nhỏ hơn ta cho chuỗi Time chứa năm đó vào $a0, chuỗi Time chứa năm lớn hơn vào $a1
- Sau đó ta lấy tháng của chuỗi $a0, và tháng của $a1 lưu vào $t0,$t1
- So sánh $t0 < $t1 thì ta trả về KC = Year(chuỗi $a1) - Year(chuỗi $a0), nếu $t0 > $t1 thì KC = Year(chuỗi $a1) - Year(chuỗi $a2) - 1
- Nếu $t0 = $t1, 2 tháng bằng nhau thì ta lấy ngày ra so sánh, nếu Day(chuỗi $a0) <= Day(chuỗi $a0) thì trả về KC = Year(chuỗi $a1) - Year(chuỗi $a0), ngược lại thì KC = Year(chuỗi $a1) - Year(chuỗi $a0) - 1.

f) int NextLeapYear(char *Time)

- Mô tả cách thức cài đặt:
  - Lấy năm($a0) từ chuỗi Time, tạo biến đếm ($t0) và năm cộng thêm một.
  - Vào vòng lặp với điều kiện biến đếm nhỏ hơn 2, sau đó trong vòng lặp ta kiểm tra năm có phải là năm nhuận hay không, nếu là năm nhuận thì gán biến đếm($t0) bằng 1 và xuất ra , ngược lại là năm không nhuận thì tăng năm($a0) thêm 1. Tiếp tục đến khi có thêm 1 năm nhuận nữa thì tăng biến đếm($t0) bằng 2 và xuất ra.

g) int KiemTraKiTu(char* chuoingay, char*chuoithang, char*chuoinam)

kết quả trả về là 0 nếu chuỗi không hợp lệ và 1 nếu chuỗi hợp lệ

- Mô tả cách thức cài đặt:
  - Vì người dùng có thể nhập vào kí tự ở các bước nhập vào ngày tháng năm nên ta phải thực hiện việc kiểm tra xem người dùng có nhập hợp lệ hay không, nếu không ta sẽ cho người dùng nhập lại, lúc nhập ngày tháng năm sẽ là nhập chuỗi kí tự sau đó sẽ kiểm tra xem chuỗi kí tự đã nhập vào có phải là kí tự số hay không (tức là kí tự thuộc đoạn 48 đến 58 trong bảng ascii).
  - Thêm 1 điều nữa là sau khi nhập chuỗi thì ta kết thúc việc nhập bằng Enter nên trong chuỗi luôn có kí tự

xuống dòng ở cuối chuỗi, ta sẽ dùng cái này để kết thúc việc kiểm tra ở từng chuỗi ngày tháng năm.

- Việc kiểm tra sẽ thực hiện như sau, đầu tiên ta xem kí tự đầu tiên của từng chuỗi ngày tháng năm có phải kí tự xuống dòng hay không, nếu phải thì đây là chuỗi rỗng => sai => trả về 0. Sau khi đã chắc chắn là cả 3 chuỗi ngày tháng năm đều không rỗng ta sẽ đi kiểm tra từng chuỗi 1, nếu có 1 chuỗi sai thì trả về 0, tất cả đúng thì trả về 1. Ta sẽ bắt đầu với việc kiểm tra ngày, load từng kí tự và kiểm tra xem nó = xuống dòng không, nếu bằng thì chuỗi ngày đúng, chuyển qua kiểm tra tháng, nếu k bằng thì xét xem nó có phải là kí tự thuộc (48,58) không, nếu không thuộc thì sai, nếu thuộc thì tiếp tục load kí tự tiếp theo và lại xét như trên. Ở chuỗi tháng và năm cũng tương tự , nếu sau khi xét cả 3 chuỗi đều đúng thì trả về kết quả là 1.
- **Ví dụ:** chuỗi ngày là "\n" đầu tiên vô sẽ kiểm tra thấy đây là chuỗi rỗng => sai. Chuỗi ngày là "1a\n" qua được bước kiểm tra rỗng, kí tự đầu tiên đúng sẽ chuyển qua kí tự thứ 2 để kiểm tra thì thấy sai => sai.Chuỗi ngày là "2\n" qua được bước kiểm tra rỗng,kí tự đầu tiên đúng, kí tự thứ 2 là xuống dòng => đúng => chuyển xuống kiểm tra kí tự tháng.
- **Sau khi kiểm tra ở hàm này thì ta sẽ biết được nhưng kí tự mà người dùng nhập vào có phải toàn là số hay không , sau bước này phải kiểm tra xem dữ liệu nhập vào có phải là 1 ngày hợp lệ không ở hàm dưới.**

h) int KiemTraSo(int ngay, int thang, int nam)

--Dữ liệu đã được kiểm tra kí tự và đã thông qua.

- Mô tả cách thức cài đặt:
  - Trước tiên ta kiểm tra Month = 2 hay không, nếu Month = 2 thì ta kiểm tra xem năm đó có phải là năm nhuận hay không, nếu đó là năm nhuận thì ta kiểm tra xem Day <= 29 hay không, nếu đúng thì ta trả về 1, nếu sai thì trả về 0. Còn không phải năm nhuận thì kiểm tra Day <= 28 hay không, đúng trả về 1, sai trả về 0
  - Nếu Month != 2 thì ta kiểm tra Month = 4 || Month = 6 | | Month = 9 || Month = 11, nếu bằng 1 trong cái trường hợp trên thì kiểm tra xem Day <= 30 hay không, nếu đúng trả về 1, sai trả về 0.

- Còn các trường hợp Month còn lại kiểm tra xem Day <= 31 hay không, đúng trả về 1, sai trả về 0.

III. Tham Khảo:

www.github.com
www.daynhauhoc.com
www.stackoverflow.com