

**ĐẠI HỌC QUỐC GIA TP.HCM**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

## KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ

## BÁO CÁO ĐỒ ÁN 2

## THƯ VIỆN TIME

Giáo viên hướng dẫn: **Lê Viết Long & Phạm Tuấn Sơn**

**Thành viên nhóm**

TRẦN QUỐC BẢO – 18120111

ĐOÀN PHÚ ĐỨC – 18120117

LÊ MINH KHOA – 18120415

**Mục lục**

[KIẾN TRÚC MÁY TÍNH VÀ HỢP NGỮ 1](#_Toc13502)

[BÁO CÁO ĐỒ ÁN 2 1](#_Toc20720)

[THƯ VIỆN TIME 1](#_Toc31050)

**[1) Thông tin thành viên và mức độ đóng góp 3](#_Toc18781)**

**[2) Đánh giá mức độ hoàn thành(%) ứng với từng yêu cầu và toàn bộ project 3](#_Toc32237)**

**[4) Cách thức và qui tắc gọi hàm 4](#_Toc12626)**

**[4.1 Lưu trữ chuỗi dữ liệu do người dùng nhập vào chuỗi TIME 4](#_Toc9622)**

**[4.2 Kiểm tra sự hợp lệ của dữ liệu đầu vào 5](#_Toc22628)**

**[4.3 Xuất chuỗi theo định dạng DD/MM/YYYY 5](#_Toc8285)**

**[4.4 Chuyển đổi định dạng chuỗi TIME 5](#_Toc14780)**

**[4.5 Cho biết ngày vừa nhập là thứ mấy trong tuần 6](#_Toc8815)**

**[4.6 Kiểm tra năm đã nhập trong chuỗi TIME có phải là năm nhuận hay không 6](#_Toc28627)**

**[4.7 Cho biết khoảng thời gian (tính theo năm) giữa hai khoảng thời gian 7](#_Toc16222)**

**[4.8 Cho biết hai năm nhuận gần với năm trong chuỗi TIME nhất 7](#_Toc32521)**

**[5) Tài liệu tham khảo: 8](#_Toc29907)**

1. **Thông tin thành viên và mức độ đóng góp**

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên** | **Mã số sinh viên** |
| **1** | Trần Quốc Bảo | 18120111 |
| **2** | Đoàn Phú Đức | 18120117 |
| **3** | Lê Minh Khoa | 18120415 |

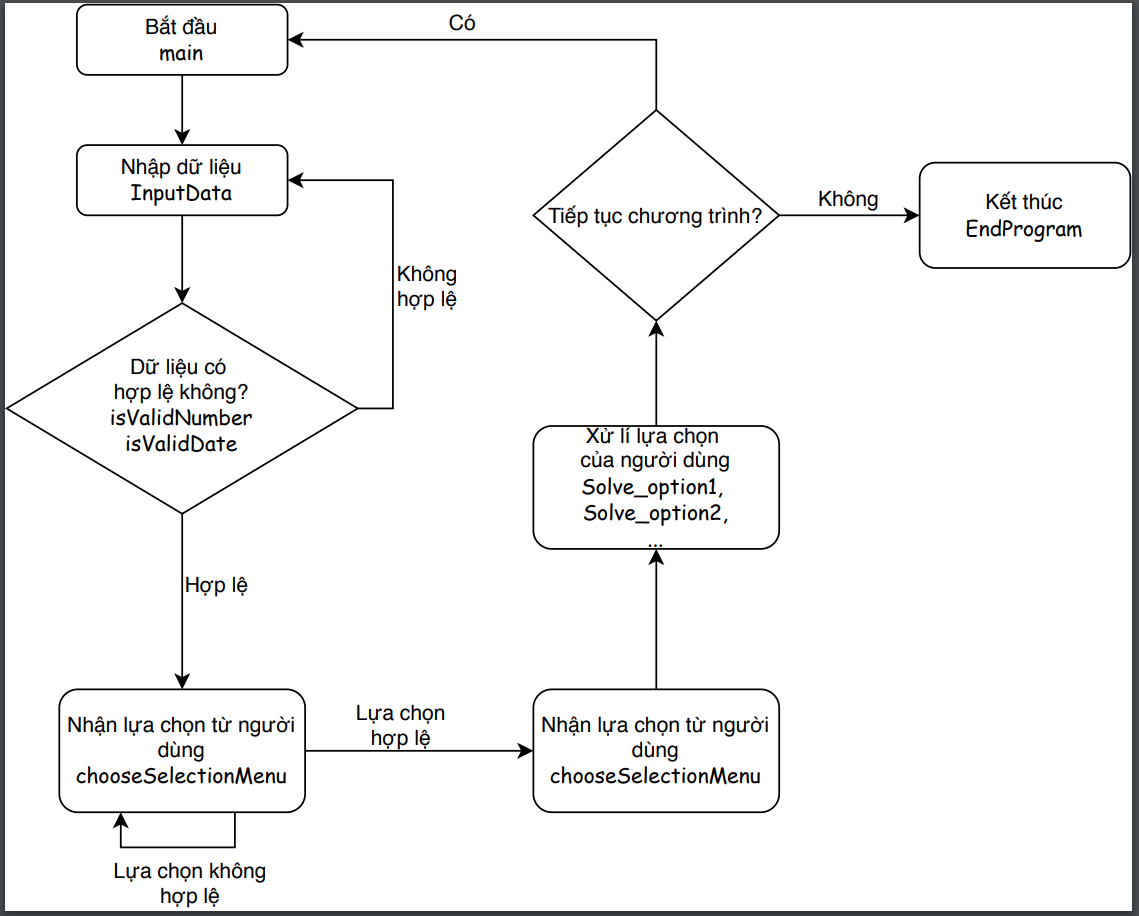
**Mức độ đóng góp đối với project:**

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên** | **Mức độ đóng góp(%)** |
| **1** | Trần Quốc Bảo | 33.33 |
| **2** | Đoàn Phú Đức | 33.33 |
| **3** | Lê Minh Khoa | 33.33 |



1. **Đánh giá mức độ hoàn thành(%) ứng với từng yêu cầu và toàn bộ project**

|  |  |
| --- | --- |
| **Các yêu cầu** | **Mức độ hoàn thành** |
| Lưu trữ chuỗi dữ liệu do người dùng nhập vào chuỗi TIME | **100%** |
| Xuất chuỗi theo định dạng DD/MM/YYYY | **100%** |
| Chuyển đổi định dạng chuỗi TIME | **100%** |
| Cho biết ngày tháng của chuỗi TIME vừa nhập | **100%** |
| Kiểm tra năm đã nhập trong chuỗi TIME có phải là năm nhuận hay không | **100%** |
| Cho biết khoảng thời gian (tính theo năm) giữa hai khoảng thời gian | **100%** |
| Cho biết hai năm nhuận gần với năm trong chuỗi TIME nhất | **100%** |
| Lấy ngày tháng năm trong chuỗi TIME | **100%** |
| Xuất ra menu để người dùng tương tác với chương trình | **100%** |
| Mức độ hoàn thành cả đồ án | **100%** |

1. **Sơ đồ các hàm xử lí giao diện menu tương tác với người dùng**
2. **Cách thức và qui tắc gọi hàm**
   1. **Lưu trữ chuỗi dữ liệu do người dùng nhập vào chuỗi TIME**

* Hàm nhập dữ liệu: inputData
  + Input: $a0 con trỏ trỏ tới chuỗi TIME
  + Output: $v0 là con trỏ trỏ tới chuỗi TIME đã được chuẩn hoá dạng DD/MM/YYYY và $v1 trả về tính hợp lệ của chuỗi.
* Chuỗi nhập không hợp lệ: nhảy lại tới inputData, yêu cầu nhập lại dữ liệu
* Chuỗi nhập hợp lệ: nhảy sang chooseSelectionMenu để in ra bảng menu.
* Sử dụng syscall 8 để nhập dữ liệu từ người dùng,
  1. **Kiểm tra sự hợp lệ của dữ liệu đầu vào**

Một dữ liệu được nhập vô phải thoả:

* isValidNumBer: nếu dữ liệu là ngày, tháng, năm, lựa chọn của người dùng thì hàm sẽ kiểm tra điều kiện các kí tự có phải là số từ 0 đến 9 hay không.
  + Input: $a0 string, không bị thay đổi trong hàm này.
  + Output: $v0 = 1 => dữ liệu hợp lệ, các kí tự là các số từ 0 đến 9. Ngược lại $v0 = 0 dữ liệu không hợp lệ.
* isValidDate: nếu dữ liệu nhập vào là chuỗi TIME, TIME\_1, TIME\_2 thì hàm sẽ kiểm tra ngày trong chuỗi này có hợp lệ hay không.
  + Input: $a0, $a1, $a2 (int) ngày tháng năm.
  + Output: $v0 = 1 => dữ liệu hợp lệ, các kí tự là các số từ 0 đến 9. Ngược lại $v0 = 0 dữ liệu không hợp lệ.
  + Chuỗi là một ngày hợp lệ nếu thoả các điều kiện:
    - 1 Tháng  12
    - 1900  Năm  9999
    - 0 < Ngày và thoả các điều kiện:
      * Nếu tháng là 1, 3, 5, 7, 8, 10, 12: ngày 31
      * Nếu tháng là 4, 6, 9, 11: ngày 30
      * Nếu là tháng 2: nếu ngày = 29 thì năm phải là năm nhuận nếu không thì ngày < 30.
  1. **Xuất chuỗi theo định dạng DD/MM/YYYY**
* Chuỗi nhập vô được lưu dưới dạng chuẩn vào chuỗi TIME
  + Ngày, tháng, năm nhập vô phải là các chuỗi số từ 0 đến 9, được lưu vào $a3 dưới dạng chuẩn DD/MM/YYYY (theo hàm inputData)
  + Hàm thực hiện là ConvertToTime:
    - Input: $a0, $a1, $a2 (int) lưu ngày tháng năm.   
      $a3 lưu chuỗi TIME
    - Output: $v0 lưu chuỗi TIME dạng chuẩn DD/MM/YYYY  
      $v1: biểu diễn sự hợp lệ (0: không hợp lệ, 1: hợp lệ)
* Sử dụng syscall 4 để xuất dữ liệu ra màn hình.
  1. **Chuyển đổi định dạng chuỗi TIME**
* Hàm thực hiện: Convert
  + Input ‘A’: goi hàm **convert\_according\_to\_A** (MM/DD/YYYY).
  + Input ‘B’: goi hàm **convert\_according\_to\_B** (Month DD, YYYY).
  + Input ‘C’: goi hàm **convert\_according\_to\_C** (DD Month, YYYY).
  + Hàm convert\_according\_to\_A: Đảo vị trí của ngày tháng trong chuỗi ban đầu.
  + Hàm convert\_according\_to\_B: Lấy tên của tháng thông qua số. Copy chuỗi DD, YYYY vào một chuỗi tạm và thực hiện nối chuỗi tạm vào chuỗi chứa tên tháng.
  + Hàm convert\_according\_to\_C: Lấy tên của tháng thông qua số. Copy chuỗi DD, YYYY vào một chuỗi tạm và thực hiện nối từng phần của chuỗi tạm vào chuỗi chứa tên tháng.
  + Input khác ba trường hợp trên, thông báo không hợp lệ bằng hàm **Convert\_invalid**
  + Output: con trỏ chứa địa chỉ chuỗi thời gian được chuyển về định dạng theo yêu cầu.
  1. **Cho biết ngày vừa nhập là thứ mấy trong tuần**
* Hàm thực hiện: Weekday
  + Input: $a0 là địa chỉ của chuỗi TIME
  + Output: $v0 thuộc tập giá trị {0, 1, 2, 3, 4, 5, 6} tương ứng với thứ trong tuần {chủ nhật, thứ 2, thứ 3, thứ 4, thứ 5, thứ 6, thứ 7}
* Nhãn hỗ trợ việc tính toán: Days\_Of\_The\_Week
  + Công thức: n = (d + 2 \* m + 3 \* (m + 1) div 5 + y div 5 + y) mod 7

Với d: ngày, m: tháng, y: năm, n: thứ

* + Nếu trường hợp đầu vào m < 3 thì :
    - m = m + 12
    - y = y – 1
  + Sau đó vẫn áp dụng công thức như trên được kết quả lưu vào $v0
  1. **Kiểm tra năm đã nhập trong chuỗi TIME có phải là năm nhuận hay không**
* Hàm thực hiện: LeapYear
  + Input: $a0 là địa chỉ chuỗi TIME
  + Output: $v0 trả về 0: không là năm nhuận, 1: là năm nhuận.
* Hàm bổ trợ: LeapYearNumber
  + Input: $a0 là năm (year)
  + Output: $v0 trả về 0: không là năm nhuận, 1: là năm nhuận.
* Các bước thực hiện:
  + Dùng hàm Year để lấy số năm ra từ chuỗi TIME ở $a0. Sau đó gán $a0 = $v0
  + Tiếp tục gọi hàm LeapYearNumber để kiểm tra $a0 có phải là năm nhuận hay không.
  + Trả về $v0 là kết quả của hàm LeapYear.
  1. **Cho biết khoảng thời gian (tính theo năm) giữa hai khoảng thời gian**
* Hàm thực hiện: GetTime
  + Input: 2 chuỗi thời gian cần tính khoảng cách giữa chúng $a0, $a1.
  + Output: khoảng cách giữa hai mốc thời gian trên (tính theo năm).
  + Các bước thực hiện:
    - Ban đầu sẽ gọi tới hàm CompareDate để so sánh hai thời gian nhập vào.
    - Dựa vào năm lớn hơn để tính khoảng cách giữa hai năm: năm của TIME\_1 trừ năm của TIME\_2 trừ 1. Nếu hai năm bằng nhau thì trả về khoảng cách bằng 0.
    - Kiểm tra xem năm của TIME\_1 có phải là năm nhuận và tháng của TIME\_1 = 2. Nếu vậy thì trừ ngày của TIME\_1 đi 1.
    - Kiểm tra xem năm của TIME\_2 có phải là năm nhuận và tháng của TIME\_2 = 2. Nếu vậy thì trừ ngày của TIME\_2 đi 1.
    - So sánh hai tháng:
      * Nếu tháng của TIME\_1 < tháng của TIME\_2 thì tăng khoảng cách lên 1. Trả về khoảng cách và kết thúc hàm.
      * Nếu tháng của TIME\_1 > tháng của TIME\_2 thì trả về khoảng cách hiện tại và kết thúc hàm.
    - So sánh hai ngày:
      * Nếu ngày của TIME\_1 <= ngày của TIME\_2 thì tăng khoảng cách lên 1. Trả về khoảng cách và kết thúc hàm.
      * Nếu ngày của TIME\_1 > ngày của TIME\_2 thì trả về khoảng cách hiện tại và kết thúc hàm
  1. **Cho biết hai năm nhuận gần với năm trong chuỗi TIME nhất**
* Hàm hỗ trợ việc tính toán: NextLeapYear
  + Input: $a0: năm hiện tại, $a1 thuộc {-1, +1} thể hiện bước nhảy lùi hay tới so với năm hiện tại
  + Output: $v0: năm tiếp theo (sau hoặc trước năm hiện tại tùy thuộc vào $a1) là năm nhuận.
* Hàm thực hiện: Leap\_Year\_Closer
  + Input: $a0 là địa chỉ của chuỗi TIME
  + Output: $v0: năm nhuận thứ nhất gần với chuỗi TIME, $v1: năm nhuận thứ 2 gần với chuỗi TIME
  + Các bước thực hiện:
    - Lấy giá trị năm trong chuỗi TIME (dùng hàm Year) lưu vào $t0
    - Lấy giá trị năm trong chuỗi TIME (dùng hàm Year) lưu vào $t1
    - Tìm năm nhuận gần nhất nhỏ hơn năm hiện tại bằng hàm NextLeapYear ($a0 = $t0 và $a1 = -1). Sau đó gán vào $t0 = $v0
    - Tìm năm nhuận gần nhất nhỏ hơn năm hiện tại bằng hàm NextLeapYear ($a0 = $t0 và $a1 = 1). Sau đó gán vào $t1 = $v0
    - Kiểm tra tính hợp lệ của $t0, $t1 (1900 <= year <= 9999)
      * Nếu cả $t0, $t1 đều hợp lệ trả về $v0 = $t0 và $v1 = $t1
      * Nếu $t0 không hợp lệ thì tính lại $t0 bằng cách
        + Tìm năm nhuận gần nhất lớn hơn $t1 bằng hàm NextLeapYear ($a0 = $t1 và $a1 = 1). Sau đó gán vào $t0 = $v0
        + Trả về $v0 = $t0 và $v1 = $t1
      * Nếu $t1 không hợp lệ thì tính lại $t1 bằng cách
        + Tìm năm nhuận gần nhất nhỏ hơn $t0 bằng hàm NextLeapYear ($a0 = $t0 và $a1 = -1). Sau đó gán vào $t1 = $v0
        + Trả về $v0 = $t0 và $v1 = $t1

1. **Tài liệu tham khảo:**

* [SYSCALL functions available in MARS](http://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html)
* <https://courses.cs.vt.edu/cs2506/Fall2014/Notes/L04.MIPSAssemblyOverview.pdf>
* Bài giảng **Kiến trúc MIPS** – ThS. Phạm Tuấn Sơn, ĐH Khoa học Tự nhiên, ĐHQG