



BÁO CÁO ĐỒ ÁN 2
XÂY DỰNG THƯ VIỆN TIME BẰNG HỢP NGỮ

Giảng viên hướng dẫn : Phạm Tuấn Sơn
Danh sách thành viên nhóm:

1. Võ Quốc Thắng - 1712162
2. Lê Nguyễn Nhật Trường - 1712195
3. Lê Tuấn Đạt - 1712329

MỤC LỤC

1. GIỚI THIỆU	3
1.1. Nội dung và yêu cầu	3
1.1.1. Nội dung	3
1.1.2. Yêu cầu	3
1.2. Phân tích đề án	3
1.2.1. Kiểu dữ liệu nhập vào	3
1.2.2. Tìm thứ trong tuần tương ứng với ngày	3
1.2.3. Tính khoảng cách giữa hai mốc thời gian	3
1.2.4. Tìm hai năm nhuận gần nhất	3
1.3. Thông tin nhóm	3
1.3.1. Thông tin thành viên	3
1.3.2. Phân công công việc	4
2. TỔNG QUAN VỀ CHƯƠNG TRÌNH	4
2.1. Sơ đồ lớp	4
2.2. Kịch bản chương trình	5
3. CÀI ĐẶT CÁC YÊU CẦU	6
3.1. Hàm Date	6
3.1.1. Mẫu hàm:	6
3.1.2. Cách xử lý:	6
3.2. Hàm Convert	6
3.2.1. Mẫu hàm:	6
3.2.2. Cách xử lý:	6
3.3. Các hàm Day, Month, Year	7
3.3.1. Các mẫu hàm:	7
3.3.2. Cách xử lý:	7
3.4. Hàm LeapYear	7
3.4.1. Mẫu hàm:	7
3.4.2. Cách xử lý:	7
3.5. Hàm GetTime	7
3.5.1. Mẫu hàm:	7
3.5.2. Cách xử lý:	8

3.6.	Hàm WeekDay	8
3.6.1.	Mẫu hàm:	8
3.6.2.	Cách xử lý:	8
3.7.	Các hàm kiểm tra dữ liệu nhập vào:	8
3.7.1.	Mẫu hàm:	8
3.7.2.	Cách xử lý:	8
4.	TỔNG KẾT	9
5.	ĐÁNH GIÁ	9
5.1.	Đánh giá đồ án	9
5.2.	Đánh giá cá nhân	9
TÀI LIỆU THAM KHẢO		10

1. GIỚI THIỆU

1.1. Nội dung và yêu cầu

1.1.1. Nội dung

Viết một chương trình minh họa có giao diện của thư viện TIME bằng hợp ngữ.

1.1.2. Yêu cầu

- Phải cài đặt đầy đủ các hàm nêu trong phần 4 dưới dạng thư viện trong MIPS.
- Các hàm thư viện không được sử dụng biến toàn cục để lưu trữ giá trị tính toán trung gian, phải sử dụng biến cục bộ.
- Tuân thủ chặt chẽ qui tắc gọi hàm, mỗi hàm vi phạm qui tắc sẽ bị trừ 50% số điểm của hàm đó.

1.2. Phân tích đề án

1.2.1. Kiểu dữ liệu nhập vào

Người dùng nhập từ bàn phím các giá trị DD, MM, YYYY tương ứng, chương trình xử lý các giá trị đầu vào như các chuỗi và tiên hành đưa vào một hàm chuyển đổi sang dạng TIME tương ứng.

1.2.2. Tìm thứ trong tuần tương ứng với ngày

Chọn ngày 01/01/0001 (chủ nhật) để làm cột mốc tính toán ra được thứ trong tuần.

1.2.3. Tính khoảng cách giữa hai mốc thời gian

Đếm khoảng cách giữa hai mốc thời gian (dựa trên cột mốc 01/01/0001), rồi cộng thêm số năm nhuận giữa hai mốc đó, tất cả chia cho 365 ngày sẽ ra được số năm khoảng cách giữa hai ngày.

1.2.4. Tìm hai năm nhuận gần nhất

Từ giá trị năm nhập vào ta sẽ tìm năm nhuận liền sau năm đó bằng cách tăng dần số năm và kiểm tra có phải là năm nhuận không. Tìm năm nhuận còn lại liền trước năm đó tương tự.

1.3. Thông tin nhóm

1.3.1. Thông tin thành viên

STT	MSSV	Họ và tên	Email	Công việc
1	1712162	Võ Quốc Thăng	voquocthang1999@gmail.com	Xử lý chức năng 4 và 6, viết báo cáo

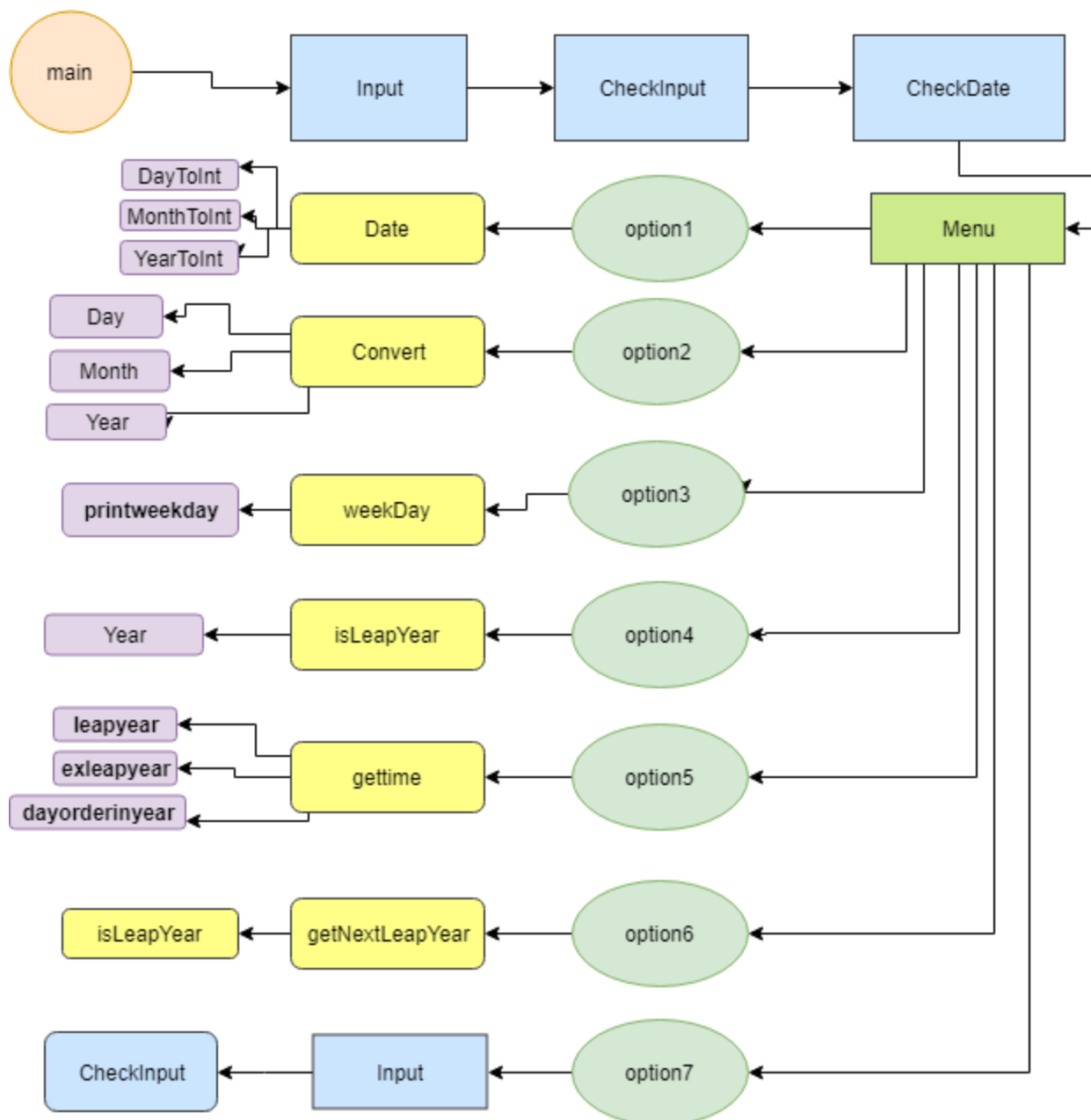
2	1712195	Lê Nguyễn Nhật Trường	truongthk62014@gmail.com	Tìm hiểu đề bài ,phân công công việc, xây dựng menu và kiểu tra chức năng số 2,7
3	1712329	Lê Tuấn Đạt	letuandat2110@gmail.com	Xử lý chức năng số 1,3 và 5, test chương trình

1.3.2. Phân công công việc

Thành viên	Công việc	Mô tả
Lê Nguyễn Nhật Trường	Tìm hiểu đề bài và phân công công việc	Dựa trên các yêu cầu trong đề để phân chia nhóm các hàm liên quan với nhau và giao việc cho thành viên thích hợp
	Xây dựng menu	Xây dựng giao diện trên cửa sổ cho người dung thực hiện thao tác
	Chức năng số 2 và số 7	Kiểm tra dữ liệu nhập vào và chuyển đổi giữa các kiểu TIME
Lê Tuấn Đạt	Chức năng số 1	Chuyển dữ liệu nhập vào về dạng DD/MM/YYYY
	Chức năng số 3	Tìm thứ của ngày nhập vào
	Chức năng số 5	Tính khoảng cách giữa hai mốc thời gian
Võ Quốc Thắng	Chức năng số 4	Kiểm tra năm đầu vào có phải năm nhuận không
	Chức năng số 6	Tìm hai năm nhuận gần nhất
	Viết báo cáo	Viết báo cáo tổng kết chương trình

2. TỔNG QUAN VỀ CHƯƠNG TRÌNH

2.1. Sơ đồ lớp



2.2. Kịch bản chương trình

- Khởi động chương trình, người dùng phải nhập vào ngày – tháng – năm theo mẫu
- (DD – MM-YYYY). Nếu nhập sai chương trình sẽ báo lỗi và yêu cầu người dùng nhập lại.
- Khi người dùng nhập theo đúng mẫu yêu cầu, chương trình sẽ thực hiện kiểm tra tính đúng đắn của dữ liệu đầu vào. Nếu nhập sai chương trình sẽ báo lỗi và yêu cầu người dùng nhập lại.

- Khi các dữ liệu nhập đều hợp lý, chương trình sẽ hiện thị Menu chức năng để người dung chọn lựa chức năng.
- Sau khi thực hiện mỗi chức năng, chương trình sẽ hỏi người dung có muốn thực hiện tiếp chương trình hay không. Nhấn số 0 nếu muốn thoát khỏi chương trình.

3. CÀI ĐẶT CÁC YÊU CẦU

3.1. Hàm Date

3.1.1. Mẫu hàm:

```
char* Date(int day, int month, int year, char* TIME)
```

3.1.2. Cách xử lý:

- Với giá trị day chỉ nằm trong đoạn [1...31] nên ta chỉ cần lấy day chia cho 10, chuyển kết quả thương thành kí tự (gọi hàm char) và chuyển số dư (gọi hàm char) rồi lưu lại trên biến TIME, ta được phần ngày (DD/_/_/_ _ _ _) trong mảng kí tự kết quả.
- Với giá trị month chỉ nằm trong đoạn [1...12] ta làm tương tự như khi s=xử lý với biến day, sau khi biến đổi ta sẽ có mảng kí tự kết quả như sau: DD/MM/_ _ _ _
- Với giá trị năm chỉ nằm trong đoạn [1...9999] nên ta phải lấy year chia lần lượt cho 1000,100,10 để nhận được các số hàng nghìn,hàng trăm,hàng chục và hàng đơn vị.Gọi hàm char để chuyển các số thành các kí tự.Sau khi lưu vào biến TIME, ta được mảng kí tự sau : DD/MM/YYYY

3.2. Hàm Convert

3.2.1. Mẫu hàm:

```
char* Convert(char* TIME, char type)
```

3.2.2. Cách xử lý:

- Tử mảng kí tự TIME: trích ra các phần DD, MM, YYYY tương ứng
- Sao chép vào các chuỗi tạm thời (dung hàm phụ Strcpy)
- Dựa trên vào kiểu nhập vào type = {typeA,typeB,typeC} để đưa ra kết quả phù hợp.
- Đối với các kiểu yêu cầu xuất ra tên tháng, chương trình đã khai báo trước một mảng (nameOfMonth) chứa các chuỗi là tên của tháng

tương ứng. Sử dụng giá trị tháng khớp với vị trí trong mảng để xuất ra tên tháng.

3.3. Các hàm Day, Month, Year

3.3.1. Các mẫu hàm:

```
int Day(char* TIME)
```

```
int Month(char* TIME)
```

```
int Year(char* TIME)
```

3.3.2. Cách xử lý:

Với mặc định kiểu TIME : DD/MM/YYYY như đề bài, việc chuyển đổi trong các hàm này cũng tương tự như trong hàm Date. Tuy nhiên ta phải đổi kiểu kí tự sang kiểu int- việc này có thể dễ dàng thực hiện bằng cách trừ giá trị cho mã ASCII 48.

3.4. Hàm LeapYear

3.4.1. Mẫu hàm:

```
int LeapYear(char* TIME)
```

3.4.2. Cách xử lý:

Lấy giá trị năm trong mốc thời gian được nhập vào (gọi hàm Year)
Sử dụng công thức sau để xác định năm đó có phải là năm nhuận hay không:

```
function isLeapYear (year):
    if ((year modulo 4 is 0) and (year modulo 100 is not 0))
    or (year modulo 400 is 0)
        then true
    else false
```

Kết quả trả về 1 nếu là năm nhuận, ngược lại trả về 0

3.5. Hàm GetTime

3.5.1. Mẫu hàm:

```
int GetTime(char* TIME_1, char* TIME_2)
```


3.5.2. Cách xử lý:

Ta sẽ đếm khoảng cách giữa các hai ngày là $time_1$ và $time_2$, rồi sau đó ta sẽ đếm số năm nhuận giữa hai ngày và cộng số năm nhuận vào và đem chia lấy dư cho 365 ta sẽ ra được số năm khoảng cách giữa hai ngày.

Ngoài ra có các hàm hỗ trợ cho quá trình chuyển đổi:

- **Int leapyear(int year):** hàm để kiểm tra năm nhuận
- **Int exleapyear(int year1, int year2):** để tính số năm nhuận giữa 2 năm
- **Int dayorderinyear(char * time):** hàm trả về ngày thứ mấy trong năm, ví dụ ngày 02/01/2019 sẽ trả về là 2.

3.6. Hàm WeekDay

3.6.1. Mẫu hàm:

```
char* Weekday(char* TIME)
```

3.6.2. Cách xử lý:

- Cột mốc là ngày 01/01/0001 là ngày chủ nhật (the world birthday)
- Ta sẽ tính khoảng cách ngày từ ngày $time$ tới ngày 01/01/0001 sau đó ta sẽ đem khoảng cách ngày ấy chia lấy dư cho 7 ta sẽ ra được các số trong $\{0,1,2,3,4,5,6\}$, sau đó ta sẽ chuyển các số này vào một hàm lấy chuỗi, dãy số trên tương ứng với $\{sun, mon, tue, wed, thur, fri, sat\}$.
- Ngoài ra còn sử dụng hàm phụ:
char* printweekday(int day): hàm có chức năng chuyển dãy số $\{0,1,2,3,4,5,6\}$ thành chuỗi tương ứng $\{sun, mon, tue, wed, thur, fri, sat\}$.

3.7. Các hàm kiểm tra dữ liệu nhập vào:

3.7.1. Mẫu hàm:

```
int CheckInput(char * TIME)
```

```
int CheckDate(char * TIME)
```

3.7.2. Cách xử lý:

- Hàm **CheckInput** có chức năng đảm bảo người dùng nhập đúng theo mẫu (DD/MM/YYYY). Nếu nhập sai (trả về 0) thì yêu cầu người dùng nhập lại, ngược lại thì gọi hàm **CheckDate** để kiểm tra tính hợp lý của ngày nhập vào.
- Hàm **CheckDate** sẽ kiểm tra các điều kiện sau:
 - Giá trị ngày phải nằm trong khoảng ngày thực tế của mỗi tháng.

- Riêng tháng hai cần phải kiểm tra xem năm đó có phải năm nhuận không để xác định số ngày trong tháng là 28 hay 29.
- Giá trị tháng có năm trong đoạn [01...12].

4. TỔNG KẾT

Trong quá trình thực hiện chương trình, nhóm đã tiếp cận sâu hơn với ngôn ngữ Hợp ngữ cũng như các sử dụng phẩm mềm MARS để lập trình MIPS. Nhóm còn hiểu hơn về cơ chế máy tính xử lý các dòng lệnh trong ngôn ngữ lập trình cấp cao hơn mà nhóm thường dung, từ đó nắm chắc hơn được các kiến thức đã học về ngôn ngữ lập trình.

5. ĐÁNH GIÁ

5.1. Đánh giá đồ án

STT	Yêu cầu	Kết quả	Mức độ hoàn thành
1	Xây dựng đầy đủ các chức năng	Hoàn thành tất cả các chức năng	100%
2	Không xử lý các biến toàn cục để lưu giá trị trung gian trong hàm	Chỉ sử dụng các thanh ghi cục bộ trong các hàm	100%
3	Tuân thủ quy tắc gọi hàm	Các tham số đầu vào đều làm theo mẫu đã mô tả	100%
4	Xây dựng menu và giao diện cho người dùng	Hoàn thành xây dựng menu cho phép người dùng nhập từ bàn phím	100%
5	Không được sử dụng các mã giả trừ la, nhân, chia	Chương trình đều chỉ chứa các câu lệnh cho phép	100%

5.2. Đánh giá cá nhân

Thành viên	Khối lượng công việc	Mức độ hoàn thành
Lê Nguyễn Nhứt Trường	35%	100%
Lê Tuấn Đạt	35%	100%
Võ Quốc Thắng	30%	100%

TÀI LIỆU THAM KHẢO

1. Slide bài giảng kiến trúc MIPS Phạm Tuấn Sơn
2. MIPS Assembly language programming, Robert Britton
3. Theo trang: <https://www.mips.com/>
4. Theo trang: https://en.wikipedia.org/wiki/MIPS_architecture
5. Theo trang: <http://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html>