

# Báo cáo Thực hành KTMT bài Giữa kỳ

Họ và tên: Nguyễn Đức Phú

MSSV: 20215116

## Bài tập phần A bài 2: Nhập số nguyên dương N từ bàn phím, in ra màn hình dãy số Fibonacci nhỏ hơn N.

### 1. Phân tích cách thực hiện:

- Bước 1: Nhập số N, báo lỗi trong trường hợp không nguyên dương
- Bước 2: Khởi tạo hai giá trị ban đầu của dãy  $F(0) = F(1) = 1$
- Bước 3: Chạy vòng lặp từ  $i=0$  có nhiệm vụ:
  - o In ra giá trị  $F(i)$  với  $i$  là chỉ số hiện tại
  - o Cộng hai giá trị  $F(i+1)$  và  $F(i)$  để có giá trị tiếp theo
  - o Gán giá trị  $F(i+1)$  vào  $F(i)$
  - o Gán giá trị tính được từ phép cộng vào  $F(i+1)$
  - o Vòng lặp dừng lại nếu giá trị đó  $> N$
- Bước 4: Khi thoát khỏi vòng lặp in giá trị  $F(i)$  cuối cùng do vòng lặp dừng trước khi nó được in ra

### 2. Kết quả thực hiện:

- Chạy toàn bộ chương trình, có thông báo lỗi trong trường hợp  $N \leq 0$

The screenshot displays a MIPS simulator interface with the following components:

- Text Segment:** A table showing assembly code with columns for Bkpt, Address, Code, Basic, and Source. The code includes instructions for loading, storing, and system calls.
- Data Segment:** A table showing memory addresses and their corresponding values in hexadecimal.
- Registers:** A table listing registers (e.g., \$zero, \$at, \$v0, \$v1) and their current values.
- Mars Messages / Run I/O:** A window showing the program's execution output, including error messages for invalid input and the resulting Fibonacci sequence.

The output in the Mars Messages window is as follows:

```
-- program is finished running (dropped off bottom) --  
  
Nhập số nguyên dương N: 0  
ERROR: Yeu cau nhap N la mot so nguyên dương. Vui long nhap lai!  
Nhập số nguyên dương N: 1234  
Dãy Fibonacci nhỏ hơn N là:  
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987  
-- program is finished running --
```

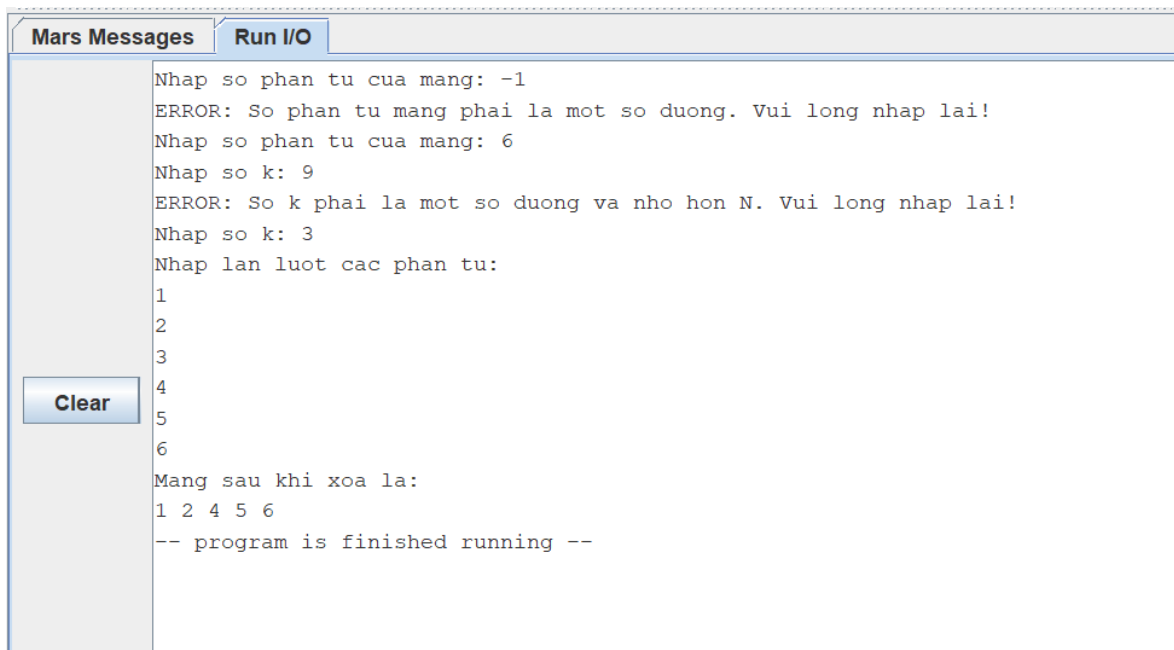
## Bài tập phần B bài 11: Nhập mảng số nguyên và số k từ bàn phím. Xóa phần tử tại vị trí k ra khỏi mảng

### 1. Phân tích cách thực hiện:

- Bước 1: Nhập số phần tử của mảng, số k và lần lượt các phần tử của mảng.  
Thông báo lỗi trong các trường hợp đặc biệt
- Bước 2: Bắt đầu chạy vòng for tại A[k-1] (do chỉ số bắt đầu từ 0) thực hiện các nhiệm vụ:
  - o Gán  $A[i] = A[i+1]$  với i là vị trí hiện tại
  - o Tăng i
  - o Thoát khỏi vòng for khi  $i = N$
  - o Trường hợp đặc biệt  $N=1$  được tách riêng
- Bước 3: Xóa bỏ phần tử cuối bằng cách gán  $= 0$

### 2. Kết quả thực hiện:

- Chạy toàn bộ chương trình, trong đó có thông báo lỗi trong các trường hợp  $N \leq 0$  và  $k > N$ :



```
Mars Messages  Run I/O
Nhap so phan tu cua mang: -1
ERROR: So phan tu mang phai la mot so duong. Vui long nhap lai!
Nhap so phan tu cua mang: 6
Nhap so k: 9
ERROR: So k phai la mot so duong va nho hon N. Vui long nhap lai!
Nhap so k: 3
Nhap lan luot cac phan tu:
1
2
3
4
5
6
Mang sau khi xoa la:
1 2 4 5 6
-- program is finished running --
```

Clear

- Mảng được lưu tại cửa sổ Data Segment như sau:

Data Segment						
Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)
0x10010000	0x00000001	0x00000002	0x00000004	0x00000005	0x00000006	0x00000000

⇒ Phần tử 3 đã bị xóa

- Với trường hợp mảng chỉ có 1 phần tử:

```

iges Run I/O
Nhap so phan tu cua mang: 1
Nhap so k: 1
Nhap lan luot cac phan tu:
1
Mang khong con phan tu nao sau khi xoa

-- program is finished running --

```

## **Bài tập phần C bài 9: Nhập vào xâu ký tự. Đổi các ký tự đầu của mỗi từ thành chữ hoa, các ký tự còn lại thành chữ thường.**

### 1. Phân tích cách thực hiện:

- Bước 1: Nhập xâu, báo lỗi nếu xâu trống
- Bước 2: Tìm độ dài xâu
- Bước 3: Khởi tạo các thanh ghi chứa các giá trị mã Ascii để tiện so sánh
- Bước 4: Chạy lần lặp đầu tiên (đặc biệt do không có dấu cách đứng trước)
- Bước 5: Chạy vòng lặp cho tất cả các ký tự còn lại
- Bước 6: In ra xâu chuẩn hóa

### • Cách chuyển đổi:

- Với chữ thường sẽ check 3 điều kiện: 2 điều kiện của mã ascii và điều kiện có dấu cách đứng trước (mã ascii nhỏ hơn hoặc bằng 32 thì là dấu cách)
  - o Sau đó trừ đi 3 để sử dụng hàm bgezal
  - o Do giá trị check chỉ tối đa là 0 (đã trừ đi 3) nên chỉ khi có chữ hoa cần chuyển đổi thì chương trình con mới được gọi
- Với chữ hoa: chỉ thay điều kiện không có dấu cách đứng trước (mã ascii lớn hơn 32 thì chắc chắn không phải dấu cách)
- Với các ký tự đặc biệt khác thì check không thể  $\geq 0$  nên được giữ nguyên

### 2. Ý nghĩa chương trình con

#### • Chương trình con `thuong_thanh_hoa`:

- Đầu vào là thanh ghi `$t9` chứa mã ascii của ký tự cần chuyển và `$a0` chứa địa chỉ của ký tự đó
- Trừ `$t9` đi 32 để lấy mã ascii của ký tự hoa tương ứng
- Lưu lại vào string
- Không có giá trị trả về

#### • Chương trình con `hoa_thanh_thuong`:

- Tương tự chương trình con ở trên chỉ thay đổi là lấy `$t9` cộng thêm 32 để lấy mã ascii của ký tự thường tương ứng

### 3. Kết quả thực hiện:

- Chạy toàn bộ chương trình, thông báo lỗi nếu xâu trống
- Các ký tự đặc biệt được giữ nguyên
- Chữ hoa và thường được chuyển đổi
- Xâu chuẩn hóa được lưu trên Data Segment

The screenshot displays a debugger interface with two main panels. The top panel, titled "Data Segment", shows a memory table with columns for Address and various Value offsets (+0, +4, +8, +c, +10, +14, +18, +1c). The first row (Address 0x10010000) contains the string "n l x o e h c c a C n a B" followed by several null bytes. The second row (Address 0x10010020) contains a series of null bytes. The third row (Address 0x10010040) contains another series of null bytes. The bottom panel, titled "Mars Messages", shows the output of a program. It includes a "Reset: reset completed." message, followed by a prompt "Nhap xau ky tu:". An error message "ERROR: Khong duoc de trong. Vui long nhap lai" is displayed. The user input "Nhap xau ky tu: x1N cH@o cAc baN" is shown. The program output "Xau chuan hoa la: X1n Ch@o Cac Ban" is displayed. The program ends with the message "-- program is finished running --". A "Clear" button is visible on the left side of the Mars Messages panel.

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	n l x	o e h c	c a C	n a B	\0 \0 \0 \n	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010020	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0
0x10010040	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0	\0 \0 \0 \0

Reset: reset completed.

Nhap xau ky tu:

ERROR: Khong duoc de trong. Vui long nhap lai

Nhap xau ky tu: x1N cH@o cAc baN

Xau chuan hoa la:

X1n Ch@o Cac Ban

-- program is finished running --