

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG  
VIỆT – NHẬT**

----- □ -----



**BÁO CÁO BÀI TẬP  
MÔN KIẾN TRÚC MÁY TÍNH  
ĐỀ TÀI SỐ 4, 8**

Giảng viên hướng dẫn: **ThS. Nguyễn Bá Vui**

Sinh viên : **Chu Văn Thành - 20194673**

**Nguyễn Trung Kiên - 20194598**

Nhóm: **15** - Lớp: **Việt Nhật K6**

# A. BÀI TẬP 8

## I. Đề Bài

Hệ thống ổ đĩa RAID5 cần tối thiểu 3 ổ đĩa cứng, trong đó phần dữ liệu parity sẽ được chứa lần lượt lên 3 ổ đĩa như trong hình bên. Hãy viết chương trình mô phỏng hoạt động của RAID 5 với 3 ổ đĩa, với giả định rằng, mỗi block dữ liệu có 4 ký tự. Giao diện như trong minh họa dưới. *Giới hạn chuỗi ký tự nhập vào có độ dài là bội của 8.*

Trong ví dụ sau, chuỗi ký tự nhập vào từ bàn phím (DCE.\*\*\*ABCD1234HUSTHUST) sẽ được chia thành các block 4 byte. Block 4 byte đầu tiên "DCE." sẽ được lưu trên Disk 1, Block 4 byte tiếp theo "\*\*\*\*" sẽ lưu trên Disk 2, dữ liệu trên Disk 3 sẽ là 4 byte parity được tính từ 2 block đầu tiên với mã ASCII là  $6e = 'D' \text{ xor } '*'$ ;  $69 = 'C' \text{ xor } '*'$ ;  $6f = 'E' \text{ xor } '*'$ ;  $04 = '.' \text{ xor } '*'$

Nhập chuỗi kí tự : DCE.***ABCD1234HUSTHUST		
Disk 1	Disk 2	Disk 3
-----	-----	-----
DCE.	****	[ 6e, 69, 6f, 04]
ABCD	[[ 70, 70, 70, 70]]	1234
[ 00, 00, 00, 00]	HUST	HUST
-----	-----	-----

## II. Định hướng

**Bước 1:** Nhập chuỗi đầu vào từ bàn phím và lưu vào bộ nhớ

**Bước 2:** Đếm độ dài chuỗi và kiểm tra độ dài chuỗi có chia hết cho 8 không. Nếu không chia hết cho 8 thì quay lại bước 1. Nếu chia hết cho 8 thì đến bước 3.

**Bước 3:** Chia chuỗi đầu vào thành các dãy 8 ký tự. Mỗi dãy 8 ký tự coi là 1 Line, 4 ký tự đầu gọi là block 1, 4 ký tự sau là block 2, và parity của dãy được gọi là block 3.

Ta phân loại được các dãy 8 ký tự thành 3 loại Line:

- Line 1: Block 1, Block2, Block3
- Line 2: Block 1, Block3, Block2
- Line 3: Block3, Block1, Block2

Các khối chỉ khác nhau về thứ tự in:

**Bước 4:** In các Line ra theo thứ tự cho đến khi hết chuỗi đầu vào

## III. Ý nghĩa của mã nguồn

1. Hàm **input**: nhập chuỗi đầu vào
2. Hàm **countLength**: đếm độ dài chuỗi đầu vào
3. Hàm **divisible\_8**: kiểm tra độ dài chuỗi có chia hết cho 8
  - Nếu chia hết cho 8 thì chạy đến hàm TinhToan
  - Nếu không chia hết cho 8 thì chạy đến hàm
4. Hàm **error\_chia8du**: In ra lỗi chuỗi đầu vào chia 8 bị dư và trở lại hàm input
5. Hàm **tinhToan**: Tính số Line cần in
6. Hàm **line**:
  - Tính toán và lưu giá trị parity vào bộ nhớ

- Phân loại Line phục vụ mục đích ra
- 7. Hàm **printLine1**, **printLine2**, **printLine3**:
  - In ra các dòng theo thứ tự được định sẵn
- 8. Hàm **congDiaChi**: tăng các biến count trong các hàm trước đó
- 9. Hàm **Hex**: chuyển giá trị thập phân sang dạng chữ số hệ 16

## IV. Mã nguồn

Mã nguồn đã được lưu và gửi cho thầy  
n08\_g14\_NguyenTrungKien.asm

## V. Hình ảnh mô phỏng

Nhập chuỗi cần lưu: DCE.\*\*\*\*ABCD1234HUSTHUST

Disk 1	Disk 2	Disk 3
DCE.	****	[[ 6e,69,6f,04]]
ABCD	[[ 70,70,70,70]]	1234
[[ 00,00,00,00]]	HUST	HUST

Nhập chuỗi cần lưu: kien2507dtmy0405

Disk 1	Disk 2	Disk 3
kien	2507	[[ 59,5c,55,59]]
dtmy	[[ 54,40,5d,4c]]	0405

-- program is finished running (dropped off bottom) --

## B. Bài tập 4

### I. Đề bài

Postscript CNC Marsbot

Máy gia công cơ khí chính xác CNC Marsbot được dùng để cắt tấm kim loại theo các đường nét được qui định trước. CNC Marsbot có một lưỡi cắt dịch chuyển trên tấm kim loại, với giả định rằng:

- Nếu lưỡi cắt dịch chuyển nhưng không cắt tấm kim loại, tức là Marsbot di chuyển nhưng không để lại vết (Track)
- Nếu lưỡi cắt dịch chuyển và cắt tấm kim loại, tức là Marsbot di chuyển và có để lại vết. Để điều khiển Marsbot cắt đúng như hình dạng mong muốn, người ta nạp vào Marsbot một mảng cấu trúc gồm 3 phần tử:
- <Góc chuyển động>, <Cắt/Không cắt>, <Thời gian>
- Trong đó <Góc chuyển động> là góc của hàm HEADING của Marsbot- <Cắt/Không cắt> thiết lập lưu vết/không lưu vết
- <Thời gian> là thời gian duy trì quá trình vận hành hiện tại

Hãy lập trình để CNC Marsbot có thể:

- Thực hiện cắt kim loại như đã mô tả
- Nội dung postscript được lưu trữ cố định bên trong mã nguồn
- Mã nguồn chứa 3 postscript và người dùng sử dụng 3 phím 0, 4, 8 trên bàn phím Key Matrix để chọn postscript nào sẽ được gia công.
- Một postscript chứa chữ DCE cần gia công. Hai script còn lại sinh viên tự đề xuất (tối thiểu 10 đường cắt)

## II. Định hướng.

B1: (Nhập từ Digital Lab Sim) người dùng sẽ nhập các số 0,4,8 vào digital lab sim để tiến hành gia công, nếu nhập không đúng sẽ phải nhập lại

B2: Tùy vào việc nhập 0,4,8 sẽ thực hiện việc gia công trên marsbot

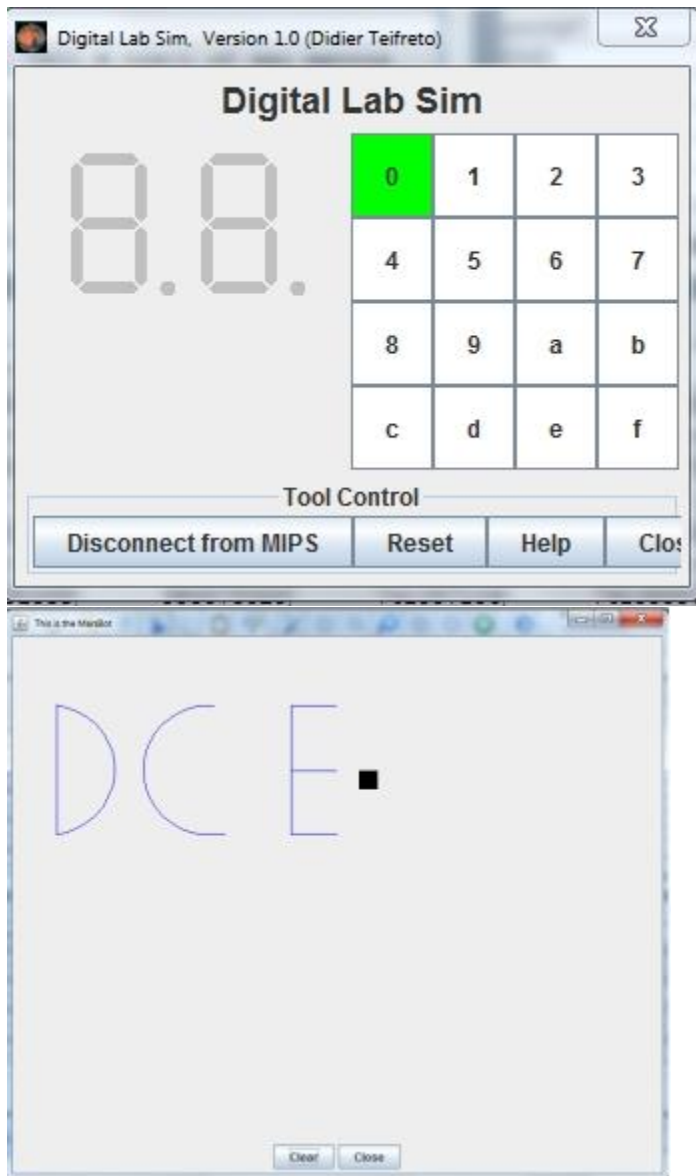
## III. Ý nghĩa mã nguồn.

- Các pscript1, pscript2, pscript3: dùng để lưu chuỗi ký tự gồm các bước để gia công trong marsbot. Trong đó
  - pscript1: các bước để gia công chữ DCE khi người dùng nhấn số 0
  - pscript2: các bước để gia công chữ THANH khi người dùng nhấn số 4
  - pscript1: các bước để gia công chữ KIEN khi người dùng nhấn số 8
  - Mỗi giá trị các bước gồm có 3 giá trị nhỏ lần lượt tượng trưng cho góc cần di chuyển đến, thời gian chờ và quyết định xem khi di chuyển có vẽ hay không( 1 là có, 0 là chỉ di chuyển chứ không vẽ)
- các hàm polling, NOT\_NUMPAD\_0, NOT\_NUMPAD\_4, COME\_BACK: kiểm tra xem người dùng đã nhập đúng các số 0,4,8 như yêu cầu chưa
- READ\_PSCRIPT: lưu các giá trị góc và thời gian chờ
- READ\_ROTATE: đọc giá trị góc
- READ\_TIME: đọc thời gian chờ
- READ\_TRACK, CHECK\_UNTRACK: kiểm tra xem giá trị cuối là 0 hay 1 để quyết định việc gia công
- INCREMENT: bỏ qua dấu ;
- GO, STOP, TRACK, UNTRACK, ROTATE, END: thực hiện việc gia công

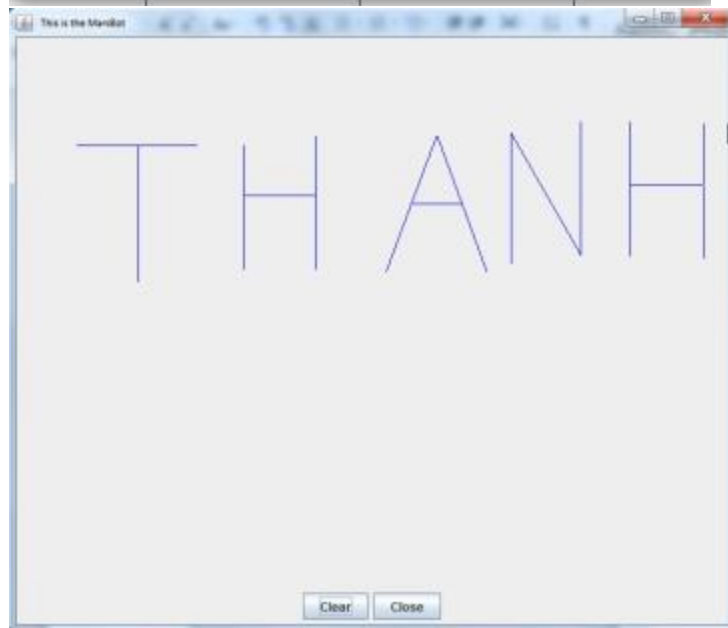
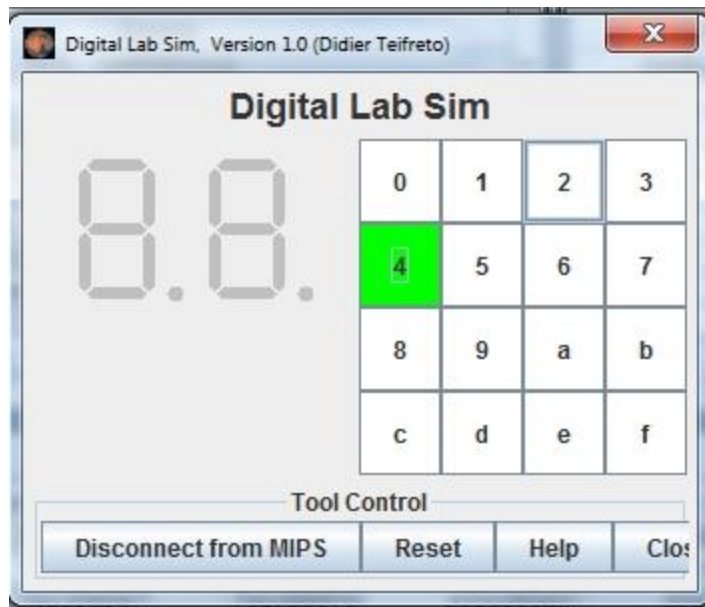
IV. Mã nguồn: đã lưu trong file

V. Hình ảnh minh họa

*Khi chọn 0*



Khi chọn 4



Khi chọn 8

