**Báo cáo Mini Projects thực hành KTMT cuối kỳ**

**Họ và tên: Đặng Hồng Minh**

**MSSV: 20225740**

**Bài 9:**

1. **Yêu cầu**

Cho hình ảnh đã được chuyển thành các kí tự ascii như hình vẽ. Đây là hình của chữ DCE có viền \* và màu là các con số.

A number and stars in a shape of a letter

Description automatically generated with medium confidence

- Hãy hiển thị hình ảnh trên lên giao diện console (hoặc giao diện Display trong công cụ giả lập Keyboard and Display MMIO Simulator)

- Hãy sửa ảnh để các chữ cái DCE chỉ còn lại viền, không còn màu số ở giữa, và hiển thị

- Hãy sửa ảnh để hoán đổi vị trí của các chữ, thành ECD, và hiển thị. Để đơn giản, các hoạt tiết đính kèm cũng được phép di chuyển theo.

- Hãy nhập từ bàn phím kí tự màu cho chữ D, C, E, rồi hiển thị ảnh trên với màu mới. Chú ý: ngoài vùng nhớ lớn chứa ảnh được chứa sẵn trong code, không được tạo thêm vùng nhớ mới để chứa ảnh hiệu chỉnh.

1. **Thuật toán thực hiện:** Dữ liệu hình ảnh sẽ được tổ chức thành 16 dòng, mỗi dòng có 62 kí tự:

• Hiển thị hình ảnh: Chức năng in ra màn hình các chữ như ban đầu sẽ in lần lượt từng dòng dữ liệu để tạo thành hình ảnh

• Hiển thị hình ảnh chỉ có viền, không có màu: Duyệt từng kí tự theo từng dòng. Nếu gặp kí tự chữ số (≥ 0 & ≤ 9, dùng để chỉ màu) thì thay kí tự đó bằng space để xóa màu. Ngược lại nếu gặp kí tự khác chữ số thì in ra như bình thường

• Hiển thị hình ảnh có thứ tự các chữ cái bị đảo: Chia vùng nhớ thành 3 phần chính tương ứng với 3 chũ cái:

* Chữ D: Từ cột 0 đến cột 21
* Chữ C: Từ cột 22 đến cột 41
* Chữ E: Từ cột 42 đến cột 59

In từng kí tự theo từng dòng lần lượt các kí tự từ vị trí 42 => 59 (chữ E), sau đó từ cột 22 => 41 (chữ C) , sau đó từ cột 0 => 21 (chữ D). Có thể tổ chức dữ liệu để có thể đảo vị trí các chữ theo thứ tự nào cũng được.

• Đổi màu: Lưu các màu hiện tại của D, C, E lần lượt vào các thanh ghi t5, t6, t7. Nhập màu muốn thay đổi lần lượt cho D, C, E và lưu vào các thanh ghi s3, s4, s5. Nếu số nhập không phải màu từ 0 => 9 thì yêu cầu nhập lại.

1. **Các thanh ghi sử dụng**

**$t0**: Thường được sử dụng làm đếm ký tự hoặc dòng.

**$t1**: Thường được sử dụng để lưu giới hạn cố định (ví dụ: độ dài của một dòng).

**$t2**: Được sử dụng để lưu trữ tạm thời các giá trị ký tự hoặc giá trị ASCII

**$t5**: Lưu trữ giá trị màu hiện tại của chữ D.

**$t6**: Lưu trữ giá trị màu hiện tại của chữ C.

**$t7**: Lưu trữ giá trị màu hiện tại của chữ E.

**$s0:** Dùng làm bộ đếm dòng hoặc chỉ số.

**$s1:** Lưu trữ tổng số dòng hoặc đóng vai trò là giới hạn vòng lặp.

**$s2:** Lưu trữ địa chỉ của chuỗi hiện tại đang được xử lý.

**$s3:** Lưu giá trị màu mới của chữ 'D'.

**$s4:** Lưu giá trị màu mới của chữ 'C'.

**$s5:** Lưu giá trị màu mới của chữ 'E'.

**$a0:** Lưu trữ địa chỉ của chuỗi cần in hoặc thông điệp nhắc nhập.

**$v0:** Được sử dụng để chứa mã syscall (ví dụ: 4 để in chuỗi, 5 để đọc số nguyên) và để lưu giữ giá trị trả về từ syscall (ví dụ: giá trị nhập từ người dùng).

**$0:** Luôn chứa giá trị 0 và được sử dụng cho các phép toán cần giá trị 0. Nó được sử dụng ngầm định trong nhiều lệnh so sánh và gán khi cần giá trị 0.

1. **Mã chương trình**

**.data**

**String1: .asciiz " \*\*\*\*\*\*\*\*\*\*\*\*\* \n"**

**String2: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*3333333333333\* \n"**

**String3: .asciiz "\*222222222222222\* \*33333\*\*\*\*\*\*\*\* \n"**

**String4: .asciiz "\*22222\*\*\*\*\*\*222222\* \*33333\* \n"**

**String5: .asciiz "\*22222\* \*22222\* \*33333\*\*\*\*\*\*\*\* \n"**

**String6: .asciiz "\*22222\* \*22222\* \*\*\*\*\*\*\*\*\*\*\*\*\* \*3333333333333\* \n"**

**String7: .asciiz "\*22222\* \*22222\* \*\*11111\*\*\*\*\*111\* \*33333\*\*\*\*\*\*\*\* \n"**

**String8: .asciiz "\*22222\* \*22222\* \*\*1111\*\* \*\* \*33333\* \n"**

**String9: .asciiz "\*22222\* \*222222\* \*1111\* \*33333\*\*\*\*\*\*\*\* \n"**

**String10: .asciiz "\*22222\*\*\*\*\*\*\*222222\* \*11111\* \*3333333333333\* \n"**

**String11: .asciiz "\*2222222222222222\* \*11111\* \*\*\*\*\*\*\*\*\*\*\*\*\* \n"**

**String12: .asciiz "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*11111\* \n"**

**String13: .asciiz " --- \*1111\*\* \n"**

**String14: .asciiz " / o o \\ \*1111\*\*\*\* \*\*\*\*\* \n"**

**String15: .asciiz " \\ > / \*\*111111\*\*\*111\* \n"**

**String16: .asciiz " ----- \*\*\*\*\*\*\*\*\*\*\* dce.hust.edu.vn \n"**

**Message0: .asciiz "------------CHON-----------\n"**

**Dong1: .asciiz "1. In chu ban dau\n"**

**Dong2: .asciiz "2. In chu khong mau\n"**

**Dong3: .asciiz "3. Doi vi tri chu\n"**

**Dong4: .asciiz "4. Doi mau chu\n"**

**Thoat: .asciiz "5. Thoat\n"**

**Nhap: .asciiz "Nhap gia tri: "**

**ChuD: .asciiz "Nhap mau cho chu D(0->9): "**

**ChuC: .asciiz "Nhap mau cho chu C(0->9): "**

**ChuE: .asciiz "Nhap mau cho chu E(0->9): "**

**.text**

**li $t5, 50 # t5 la mau chu hien tai cua chu D**

**li $t6, 49 # t6 la mau chu hien tai cua chu C**

**li $t7, 51 # t7 la mau chu hien tai cua chu E**

**main: la $a0, Message0 # nhap menu**

**li $v0, 4**

**syscall**

**la $a0, Dong1**

**li $v0, 4**

**syscall**

**la $a0, Dong2**

**li $v0, 4**

**syscall**

**la $a0, Dong3**

**li $v0, 4**

**syscall**

**la $a0, Dong4**

**li $v0, 4**

**syscall**

**la $a0, Thoat**

**li $v0, 4**

**syscall**

**la $a0, Nhap**

**li $v0, 4**

**syscall**

**li $v0, 5**

**syscall**

**Case1menu:**

**addi $v1, $0, 1**

**bne $v0, $v1, Case2menu**

**j Menu1**

**Case2menu:**

**addi $v1, $0, 2**

**bne $v0, $v1, Case3menu**

**j Menu2**

**Case3menu:**

**addi $v1, $0, 3**

**bne $v0, $v1, Case4menu**

**j Menu3**

**Case4menu:**

**addi $v1, $0, 4**

**bne $v0, $v1, Case5menu**

**j Menu4**

**Case5menu:**

**addi $v1, $0, 5**

**bne $v0, $v1, Defaultmenu**

**j Exit**

**Defaultmenu:**

**j main**

**# in ra chu ban dau**

**Menu1:**

**addi $t0, $0, 0 # bien dem = 0**

**addi $t1, $0, 16**

**la $a0, String1**

**Loop: beq $t1, $t0, main**

**li $v0, 4**

**syscall**

**addi $a0, $a0, 62**

**addi $t0, $t0, 1**

**j Loop**

**# xoa mau o giua chi giu lai vien**

**Menu2: addi $s0, $0, 0 # bien dem tung hang = 0**

**addi $s1, $0, 16**

**la $s2, String1 # $s2 la dia chi cua string1**

**Lap1: beq $s1, $s0, main**

**addi $t0, $0, 0 # $t0 la bien dem tung ki tu cua 1 hang = 0**

**addi $t1, $0, 62 # $t1 max 1 hang la 62 ki tu**

**In1hang:**

**beq $t1, $t0, End**

**lb $t2, 0($s2) # $t2 luu gia tri cua tung phan tu trong string1**

**bgt $t2, 47, Lonhon0# neu lon hon 0 thi nhay den Lonhon0**

**j Intungkitu**

**Lonhon0: bgt $t2, 57, Intungkitu # neu lon hon 9 nua thi van ko doi**

**addi $t2, $0, 0x20 # thay doi $t2 thanh dau cach**

**j Intungkitu**

**Intungkitu: li $v0, 11 # in tung ki tu**

**addi $a0, $t2, 0**

**syscall**

**addi $s2, $s2, 1# sang chu tiep theo**

**addi $t0, $t0, 1# bien dem chu**

**j In1hang**

**End: addi $s0, $s0, 1 # tang bien dem hang len 1**

**j Lap1**

**# doi vi tri chu**

**Menu3: addi $s0, $0, 0 # bien dem tung hang = 0**

**addi $s1, $0, 16**

**la $s2, String1 # $s2 luu dia chi cua string1**

**Lap2: beq $s1, $s0, main**

**# tao thanh 3 string nho**

**sb $0, 21($s2)**

**sb $0, 41($s2)**

**sb $0, 59($s2)**

**# doi vi tri**

**li $v0, 4**

**la $a0, 42($s2) # in chu E**

**syscall**

**li $v0, 4**

**la $a0, 0($s2) # in chu D**

**syscall**

**li $v0, 4**

**la $a0, 22($s2) # in chu C**

**syscall**

**li $v0, 4**

**la $a0, 60($s2)**

**syscall**

**# ghep lai thanh string ban dau**

**addi $t1, $0, 0x20**

**sb $t1, 21($s2)**

**sb $t1, 41($s2)**

**sb $t1, 59($s2)**

**addi $s0, $s0, 1**

**addi $s2, $s2, 62**

**j Lap2**

**# doi mau cho chu**

**Menu4:**

**NhapmauD: li $v0, 4**

**la $a0, ChuD**

**syscall**

**li $v0, 5 # lay mau cua ki tu D**

**syscall**

**blt $v0, 0, NhapmauD**

**bgt $v0, 9, NhapmauD**

**addi $s3, $v0, 48 # $s3 luu mau cua chu D**

**NhapmauC: li $v0, 4**

**la $a0, ChuC**

**syscall**

**li $v0, 5 # lay mau cua ki tu C**

**syscall**

**blt $v0, 0, NhapmauC**

**bgt $v0, 9, NhapmauC**

**addi $s4, $v0, 48 # $s4 luu mau cua chu C**

**NhapmauE: li $v0, 4**

**la $a0, ChuE**

**syscall**

**li $v0, 5 # lay mau cua ki tu E**

**syscall**

**blt $v0, 0, NhapmauE**

**bgt $v0, 9, NhapmauE**

**addi $s5, $v0, 48 # $s5 luu mau cua chu E**

**addi $s0, $0, 0 # bien dem tung hang =0**

**addi $s1, $0, 16**

**la $s2, String1 # $s2 la dia chi cua string1**

**li $a1, 48 # gia tri cua so 0**

**li $a2, 57 # gia tri cua so 9**

**Lapdoimau: beq $s1, $s0, Updatemau**

**addi $t0, $0, 0 # $t0 la bien dem tung ki tu cua 1 hang = 0**

**addi $t1, $0, 62 # $t1 max 10. hang la 62 ki tu**

**In1hangdoimau:**

**beq $t1, $t0, Enddoimau**

**lb $t2, 0($s2) # $t2 luu gia tri cua tung phan tu trong string1**

**CheckD: bgt $t0, 21, CheckC # kiem tra het chu D chua**

**beq $t2, $t5, fixD**

**j Tmpdoimau**

**CheckC: bgt $t0, 41, CheckE # kiem tra het chu E chua**

**beq $t2, $t6, fixC**

**j Tmpdoimau**

**CheckE: beq $t2, $t7, fixE**

**j Tmpdoimau**

**fixD: sb $s3 0($s2)**

**j Tmpdoimau**

**fixC: sb $s4 0($s2)**

**j Tmpdoimau**

**fixE: sb $s5 0($s2)**

**j Tmpdoimau**

**Tmpdoimau: addi $s2, $s2, 1 # sang chu tiep theo**

**addi $t0, $t0, 1 # bien dem chu**

**j In1hangdoimau**

**Enddoimau: li $v0, 4**

**addi $a0, $s2, -62**

**syscall**

**addi $s0, $s0, 1 # tang bien dem hang len 1**

**j Lapdoimau**

**Updatemau: move $t5, $s3**

**move $t6, $s4**

**move $t7, $s5**

**j main**

**Exit:**

1. **Kết quả chạy chương trình**

- In chữ ban đầu: **A screen shot of a computer code

Description automatically generated**

- In chữ không màu: **A white background with black text and small black letters

Description automatically generated with medium confidence**

- Đổi vị trí các chữ: **A computer screen shot of a number

Description automatically generated**

**-** Đổi màu chữ: **A screen shot of a computer code

Description automatically generated**