

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



BÀI TẬP LỚN
THỰC HÀNH
KIẾN TRÚC MÁY TÍNH

Giảng viên hướng dẫn: Hoàng Văn Hiệp

Lớp: 107349

Nhóm sinh viên thực hiện: Phạm Ngọc Bá - 20160300

Phùng Thành Công - 20160500

Hà Nội - 2019

Mục lục

LỜI MỞ ĐẦU	3
BÀI 2.....	4
<i>Đề bài.....</i>	<i>4</i>
<i>Phân tích yêu cầu</i>	<i>4</i>
<i>Cách làm</i>	<i>4</i>
<i>Mã nguồn.....</i>	<i>5</i>
<i>Hình ảnh kết quả.....</i>	<i>11</i>
BÀI 9.....	12
<i>Đề bài.....</i>	<i>12</i>
<i>Phân tích thuật toán, cách thực hiện.....</i>	<i>12</i>
<i>Mã nguồn.....</i>	<i>13</i>
<i>Hình ảnh kết quả.....</i>	<i>17</i>

Lời nói đầu

Nhóm gồm 2 thành viên Phạm Ngọc Bá và Phùng Thành Công.

Bản báo cáo khái quát quá trình thực hiện 2 bài tập lớn là bài 2 và bài 9

Gồm các nội dung chính:

- ❖ Đề bài
- ❖ Phân tích đề bài
- ❖ Cách làm
- ❖ Mã nguồn
- ❖ Hình ảnh kết quả

Bản báo cáo sẽ không tránh khỏi những sai sót. Nhóm rất mong nhận được ý kiến góp ý của thầy giáo và các bạn.

Chúng em chân thành cảm ơn.

➤ Bài 2: Vẽ hình trên màn hình Bitmaps

Sinh viên thực hiện: Phùng Thành Công

Đề bài:

Viết một chương trình sử dụng MIPS để vẽ một quả bóng di chuyển trên màn hình mô phỏng Bitmap của Mars). Nếu đối tượng đập vào cạnh của màn hình thì sẽ di chuyển theo chiều ngược lại.

Yêu cầu:

- Thiết lập màn hình ở kích thước 512x512. Kích thước 1 pixel 1x1.
- Quả bóng là một đường tròn

Chiều di chuyển phụ thuộc vào phím người dùng bấm, gồm có (di chuyển lên (W), di chuyển xuống (S), Sang trái (A), Sang phải (D) trong bộ giả lập Keyboard and Display MMIO Simulator). Tốc độ bóng di chuyển là không đổi. Vị trí bóng ban đầu ở giữa màn hình.

1. Phân tích yêu cầu:

- Vẽ một hình tròn ở giữa màn hình có tâm O(256,256), R=24.
- Sử dụng các nút bấm di chuyển lên (W), di chuyển xuống (S), Sang trái (A), Sang phải (D) để đổi hướng di chuyển của quả bóng.
- Khi quả bóng va vào thành thì bị bật ngược lại.

2. Cách làm:

- Sử dụng một mảng dữ liệu để lưu tọa độ các điểm của đường tròn.
- Mã C:

```
i = 0;
int A[];
For(i = 0 ; i < R ; i++){
    j = i^2;
    j = sqrt(R^2 - j^2);
    A[i] = j;
    A[i++];
}
```

- Sử dụng 1 vòng lặp for để vẽ đường tròn. Tạo 8 điểm trên đường tròn : (Xo + i, Yo + j), (Xo + j, Yo + i), (Xo + i, Yo - j), (Xo + j, Yo - i), (Xo - i, Yo - j), (Xo - j, Yo - i), (Xo - i, Yo + j),

(Xo - j, Yo + i). Cho i chạy từ 0 đến 24, lấy các giá trị lần lượt từ mảng 24 phần tử đã tạo ở trên. Vòng lặp chạy sẽ tạo đường tròn.

- Lưu các nút bằng các kí tự trong bảng mã ASCII vào 1 thanh ghi. Sau khi ấn phím lên xuống trái phải, sẽ kiểm tra từng trường hợp một. ấn phím Enter hình tròn sẽ chuyển về màu đen trùng với màu màn hình. Chương trình kết thúc.

3. Mã nguồn:

```
.eqv SCREEN 0x10010000 #Man hình bitmap
.eqv RED 0x00FF0000
.eqv BACKGROUND 0x00000000
.eqv KEY_A 0x00000061
.eqv KEY_S 0x00000073
.eqv KEY_D 0x00000064
.eqv KEY_W 0x00000077
.eqv KEY_ENTER 0x0000000a
.eqv DELTA_X 10
.eqv DELTA_Y 10
.eqv DELAY_TIME 100
.eqv KEY_CODE 0xFFFF0004 #Ki tu go vao
.eqv KEY_READY 0xFFFF0000 #Kiem ki tu da san sang de doc chua

# Ham delay
# Dung he thong 1 khoang thoi gian DELAY_TIME ( tinh bang mili giay)
.macro delay
    li $a0, DELAY_TIME
    li $v0, 32
    syscall
.end_macro
.macro branchIfLessOrEqual(%r1, %r2, %branch)
    sle $v0, %r1, %r2 # moveToLeft
    bnez $v0, %branch
.end_macro
.macro setColorAndDrawCircle(%color)
    li $s5, %color # Dat mau
    jal drawCircle # de xoa duong tron cu.
.end_macro

.macro addToStack3(%r1, %r2, %r3)
```

```

        add $sp, $sp, -12
        sw %r1, 0($sp)
        sw %r2, 4($sp)
        sw %r3, 8($sp)
    .end_macro
    .macro addToStack4(%r1, %r2, %r3, %r4)
        add $sp, $sp, -16
        sw %r1, 0($sp)
        sw %r2, 4($sp)
        sw %r3, 8($sp)
        sw %r4, 12($sp)
    .end_macro
    .macro getFromStack3(%r1, %r2, %r3)
        lw %r1, 0($sp)
        lw %r2, 4($sp)
        lw %r3, 8($sp)
        add $sp, $sp, 12
    .end_macro
    .macro getFromStack4(%r1, %r2, %r3, %r4)
        lw %r1, 0($sp)
        lw %r2, 4($sp)
        lw %r3, 8($sp)
        lw %r4, 12($sp)
        add $sp, $sp, 16
    .end_macro
    .kdata
        CIRCLE_DATA: .space 512
    .text
    li $s0, 256      # Xo = 256           Toa do X cua tam duong tron
    li $s1, 256      # Yo = 256           Toa do Y cua tam duong tron
    li $s2, 24       # R = 24             Ban kinh cua tam duong tron
    li $s3, 512      # SCREEN_WITH = 512   Be ngang man hinh
    li $s4, 512      # SCREEN_HEIGHT = 512 Chieu cao man hinh
    li $s5, RED      #                   Dat mau hinh tron la mau van
    mul $s6, $s2, $s2 #                   R^2
    li $s7, 0        #                   dx = 0
    li $t8, DELTA_Y  #                   dy = 10
    jal circleInit

gameLoop:
readKeyboard:
    lw $k1, KEY_READY #Vong lap cho ban phim san sang
    beqz $k1, positionCheck

```

```

    nop
    lw $k0, KEY_CODE      #Doc ky tu ban phim
    beq $k0, KEY_A, case_a #Kiem tra nut A
    beq $k0, KEY_S, case_s #Kiem tra nut S
    beq $k0, KEY_D, case_d #Kiem tra nut D
    beq $k0, KEY_W, case_w      #Kiem tra nut W
    beq $k0, KEY_ENTER, case_enter #Kiem tra nut ENTER
    j positionCheck
case_a:
    jal moveToLeft
    j positionCheck
case_s:
    jal moveToDown
    j positionCheck
case_d:
    jal moveToRight
    j positionCheck
case_w:
    jal moveToUp
    j positionCheck
case_enter:
    j endProgram

positionCheck:
checkRightEdge:
    add $v0, $s0, $s2      #
    add $v0, $v0, $s7      # If Xo + R + DELTA_X > SCREEN_WIDTH Then
    branchIfLessOrEqual($v0, $s3, checkLeftEdge) #      moveToLeft
    jal moveToLeft #
    nop
checkLeftEdge:
    sub $v0, $s0, $s2      #
    add $v0, $v0, $s7      # # If Xo - R + DELTA_X < 0 then moveToRight
    branchIfLessOrEqual($zero, $v0, checkTopEdge)
    jal moveToRight #
    nop
checkTopEdge:
    sub $v0, $s1, $s2      #
    add $v0, $v0, $t8      #
    branchIfLessOrEqual($zero, $v0, checkBottomEdge) # If Yo - R + DELTA_Y < 0 then
    moveToDown
    jal moveToDown      #
    nop

```

checkBottomEdge:

```
    add $v0, $s1, $s2      #
    add $v0, $v0, $t8      #
    branchIfLessOrEqual($v0, $s4, draw)  # If  $Y_o + R + \Delta Y > \text{SCREEN\_HEIGHT}$  then
    jal moveToUp          #
    nop
```

draw:

```
    setColorAndDrawCirle(BACKGROUND) # Ve duong tron trung mau nen
    add $s0, $s0, $s7      #      Cap nhat toa do moi
    add $s1, $s1, $t8      #      cua duong tron

    setColorAndDrawCirle(RED) # Ve duong tron moi
    delay          #      Dung chuong trinh 1 khoang
    j gameLoop
```

endProgram:

```
    setColorAndDrawCirle(BACKGROUND)
    li $v0, 10
    syscall
```

#

Ham khoi dong duong tron

Tao mang du lieu luu toa do cac diem cua duong tron

circleInit:

```
    addToStack4($ra, $t0, $t3, $t5)
```

```
    li $t0, 0          #  $i = 0$ 
```

```
    la $t5, CIRCLE_DATA
```

loop: slt \$v0, \$t0, \$s2 # $i \rightarrow R$

```
    beqz $v0, end_circleInit
```

```
    mul $t3, $t0, $t0      #  $i^2$ 
```

```
    sub $t3, $s6, $t3      #  $t3 = R^2 - i^2$ 
```

```
    move $v0, $t3
```

```
    jal sqrt
```

```
    sw $a0, 0($t5)         # Luu  $j = \sqrt{R^2 - i^2}$  vao mang du lieu
```

```
    addi $t0, $t0, 1
```

```
    add $t5, $t5, 4
```



```

        j loop
end_circleInit:
    getFromStack4($ra, $t0, $t3, $t5)
    jr $ra

#
#   Ham ve diem tren duong tron
#   Ve dong thoi 2 diem (X0 + i, Y0 + j ) va (Xo + j, Xo + i)
#   @param i = $a0, j = $a1
drawCirclePoint:
    addToStack3($t1, $t2, $t4)

    add $t1, $s0, $a0      # Xi = Xo + i
    add $t4, $s1, $a1      # Yi = Y0 + j
    mul $t2, $t4, $s3      # Yi * SCREEN_WITH
    add $t1, $t1, $t2      # Yi * SCREEN_WITH + Xi (Toa do 1 chieu cua diem anh)
    sll $t1, $t1, 2        # Dia chi tuong doi cua diem anh
    sw $s5, SCREEN($t1)    # Draw anh
    add $t1, $s0, $a1      # Xi = Xo + j
    add $t4, $s1, $a0      # Yi = Y0 + i
    mul $t2, $t4, $s3      # Yi * SCREEN_WITH
    add $t1, $t1, $t2      # Yi * SCREEN_WITH + Xi (Toa do 1 chieu cua diem anh)
    sll $t1, $t1, 2        # Dia chi tuong doi cua diem anh
    sw $s5, SCREEN($t1)    # Draw anh

    getFromStack3($t1, $t2, $t4)
    jr $ra

#   Ham ve duong tron
#   Su dung du lieu o mang CIRCLE_DATA tao boi ham khoidong_Duong tron
#Ve tung ¼ cua duong tron
drawCircle:

    addToStack4($ra, $t0, $t1, $t3)
    li $t0, 0 # init i = 0
loop_drawCircle:
    slt $v0, $t0, $s2
    beqz $v0, end_drawCircle

    sll $t1, $t0, 2
    lw $t3, CIRCLE_DATA($t1) # Load j to $t3

```

```

    move $a0, $t0    # $a0 = i
    move $a1, $t3    # $a1 = j
    jal drawCirclePoint    # Draw (Xo + i, Yo + j), (Xo + j, Yo + i)
    sub $a1, $zero, $t3
    jal drawCirclePoint    # Draw (Xo + i, Yo - j), (Xo + j, Yo - i)
    sub $a0, $zero, $t0
    jal drawCirclePoint    # Draw (Xo - i, Yo - j), (Xo - j, Yo - i)
    add $a1, $zero, $t3
    jal drawCirclePoint    # Draw (Xo - i, Yo + j), (Xo - j, Yo + i)

    addi $t0, $t0, 1
    j loop_drawCircle
end_drawCircle:

    getFromStack4($ra, $t0, $t1, $t3)
    jr $ra

```

Ham tinh can bac hai

```

# $v0 = S, $a0 = sqrt(S)
sqrt:
    mtc1 $v0, $f0
    cvt.s.w $f0, $f0
    sqrt.s $f0, $f0
    cvt.w.s $f0, $f0
    mfc1 $a0, $f0
    jr $ra

```

```

# Cac ham di chuyen
moveToLeft:
    li $s7, -DELTA_X
    li $t8, 0
    jr $ra
moveToRight:
    li $s7, DELTA_X
    li $t8, 0
    jr $ra

```

moveToUp:

```
li $s7, 0
```

```
li $t8, -DELTA_Y
```

```
jr $ra
```

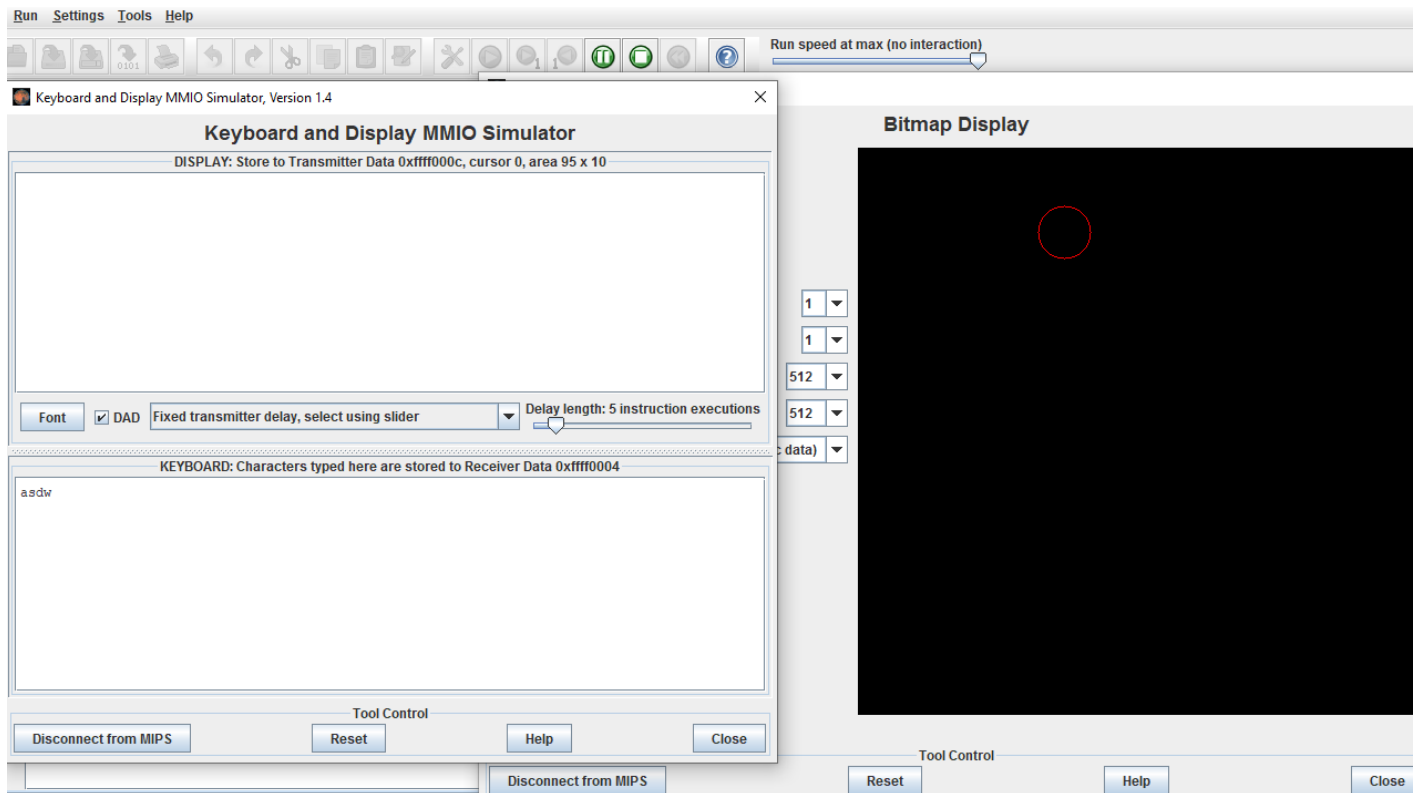
moveToDown:

```
li $s7, 0
```

```
li $t8, DELTA_Y
```

```
jr $ra
```

4. Hình ảnh kết quả mô phỏng:



Hình ảnh mô phỏng khi gõ phím W

➤ Bài 9: Vẽ hình bằng ký tự ASCII

Sinh viên thực hiện: Phạm Ngọc Bá

1. Đề bài: Cho hình ảnh đã được chuyển thành các ký tự ascii như hình vẽ. Đây là hình của chữ DCE có viền * và màu là các con số.

```
*****
*222222222222222*
*22222*****22222*
*22222*      *22222*
*22222*      *22222*      *****
*22222*      *22222*      **11111*****111*
*22222*      *22222*      **1111**      **
*22222*      *222222*      *1111*
*22222*****222222*      *11111*
*2222222222222222*      *11111*
*****      *11111*
      ---      *1111**
      /  o  o  \      *1111****  *****
      \  >  /      **111111****111*
      -----      *****
                        dce.hust.edu.vn
```

- 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 ký 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.*

2. Phân tích thuật toán, cách thực hiện

- Yêu cầu 1: Hiển thị hình ảnh chữ DCE trên giao diện console.
 - Tạo các String có cùng độ dài 64 gồm các ký tự (chữ số, dấu *, dấu cách) để tạo hình DCE như yêu cầu đề bài.
 - Sử dụng vòng lặp và lệnh syscall để in ra các dòng String ra màn hình console.
- Yêu cầu 2: Sửa hình ảnh để chữ DCE chỉ còn phần viền và hiển thị
 - Sử dụng vòng lặp cho từng dòng String, thực hiện lấy địa chỉ lưu vào \$s2.
 - Dùng vòng lặp, tăng dần giá trị \$s2 để lấy từng ký tự trong String.
 - Dùng lệnh bgt để kiểm tra ký tự đó:

- + Nếu ký tự đó nằm ngoài khoảng 48 đến 57 (tức ký tự 0 đến ký tự 9 trong bảng ASCII) thì nó được giữ nguyên.
 - + Nếu nằm trong khoảng này thì nó sẽ được thay thế bằng dấu cách (hexa: 0x20)
 - In từng ký tự đó ra màn hình.
- c) Yêu cầu 3: Hoán đổi vị trí DCE thành ECD và hiển thị.
- Lấy địa chỉ của từng chuỗi lưu vào \$s2.
 - Để hoán đổi vị trí ta xác định vị trí của từng chữ D, C, E trong chuỗi Stringi.
 - + D: từ 0 đến 22.
 - + C: từ 23 đến 42.
 - + E: từ 43 đến 61
 - Dùng lệnh sb để thêm '\0' vào mỗi vị trí 22, 42, 61 trong chuỗi.
 - In ra màn hình từng chuỗi con của Stringi theo thứ tự 43(\$s2), 23(\$s2), 0(\$s2).
 - Trả lại chuỗi Stringi ban đầu bằng cách thay thế '\0' bằng dấu cách (hexa 0x20)
- d) Yêu cầu 4: Nhập màu cho từng chữ D, C, E từ bàn phím và hiển thị DCE với màu mới.
- Giá trị màu mặc định của D, C, E được lưu vào \$t5, \$t6, \$t7.
 - Nhập giá trị màu mới cho D, C, E lưu vào \$s3, \$s4, \$s5.
 - Làm tương tự Yêu cầu 2: ký tự nào trong chuỗi có giá trị bằng \$t5 sẽ được thay bằng giá trị của \$t3, giá trị bằng \$t6 thay bằng \$s4, giá trị bằng \$t7 thay bằng \$s5. Sau khi được thay thế, in chuỗi ra màn hình.
 - Cập nhập giá trị màu mặc định của các chữ D, C, E.

3. Mã nguồn

.data

```
String1: .ascii "*****\n"
String2: .ascii "333333333333*\n"
String3: .ascii "*222222222222*\n"
String4: .ascii "*22222*****22222*\n"
String5: .ascii "*22222*22222*\n"
String6: .ascii "*22222*22222*\n"
String7: .ascii "*22222*22222*111111111111*\n"
String8: .ascii "*22222*22222*11111*\n"
String9: .ascii "*22222*22222*1111*\n"
String10: .ascii "*22222*****22222*11111*\n"
String11: .ascii "*2222222222222222*11111*\n"
String12: .ascii "*****11111*\n"
String13: .ascii "----11111*\n"
String14: .ascii " / o o | 111111111111*\n"
String15: .ascii " | > / 111111111111*\n"
String16: .ascii "-----dce.hust.edu.vn\n"
Message0: .ascii "-----MENU-----\n"
Phan1: .ascii "1. In chu\n"
Phan2: .ascii "2. In DCE chi co vien *\n"
Phan3: .ascii "3. Doi vi tri DCE -> ECD\n"
Phan4: .ascii "4. Thay doi mau chu\n"
Thoat: .ascii "5. Thoat\n"
```

```

    Nhap:          .asciiiz"Nhap gia tri: "
    ChuD:          .asciiiz"Nhap màu cho chu D(0->9): "
    ChuC:          .asciiiz"Nhap màu cho chu C(0->9): "
    ChuE:          .asciiiz"Nhap màu cho chu E(0->9): "
.text
#####
    li $t5 50 #t5 mau chu hien tai cua chu D
    li $t6 49 #t6 mau chu hien tai cua chu C
    li $t7 51 #t7 mau chu hien tai cua chu E
#####
main:
    la $a0, Message0    # nhap menu
    li $v0, 4
    syscall

    la $a0, Phan1
    li $v0, 4
    syscall
    la $a0, Phan2
    li $v0, 4
    syscall
    la $a0, Phan3
    li $v0, 4
    syscall
    la $a0, Phan4
    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 #####

```

```

Menu1:
    addi $t0, $0, 0      #bien dem dong =0
    addi $t1, $0, 16     #Tong so dong String =16
    la $a0,String1
Loop:   beq $t1, $t0, main
        li $v0, 4
        syscall

        addi $a0, $a0, 64 #Tong so ky tu trong 1 dong String la 64
        addi $t0, $t0, 1
        j Loop

##### bo het so o giua chi giu lai vien#####
Menu2:   addi $s0, $0, 0  #bien dem tung hàng =0
        addi $s1, $0, 16
        la $s2,String1 # $s2 la dia chi cua string1

Lap:     beq $s1, $s0, main
        addi $t0, $0, 0  # $t0 la bien dem tung kí tu cua 1 hàng =0
        addi $t1, $0, 64 # $t1 max 1 hàng là 64 kí tu

Inlhang:
    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 Tmp
Lonhon0: bgt $t2, 57, Tmp #neu lon hon 9 nua thi van ko doi
        addi $t2 $0 0x20 # thay doi $t2 thanh dau cach
Tmp:     li $v0, 11 # in tung kí tu
        addi $a0 $t2 0
        syscall

        addi $s2 $s2 1 #sang chu tiep theo
        addi $t0, $t0, 1# bien dem chu
        j Inlhang
End:     addi $s0 $s0 1 # tang bien dem hàng lên 1
        j Lap

#####doi vi tri chu #####
Menu3:   addi $s0, $0, 0  #bien dem tung hàng =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 $0 22($s2) #chu D
        sb $0 42($s2) #chu C
        sb $0 61($s2) #chu E
        #doi vi tri
        li $v0, 4
        la $a0 43($s2) #in chu E
        syscall

        li $v0, 4
        la $a0 23($s2) # in chu C
        syscall

        li $v0, 4
        la $a0 0($s2) # in chu D
        syscall

```

```

li $v0, 4
la $a0 62($s2)
syscall
# ghep lai thanh string ban dau
addi $t1 $0 0x20
sb $t1 22($s2)
sb $t1 42($s2)
sb $t1 61($s2)

addi $s0 $s0 1
addi $s2 $s2 64
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 hàng =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 hàng =0
            addi $t1, $0, 64      # $t1 max 1 hàng là 64 kí tu

Inlhangdoimau:
            beq $t1, $t0, Enddoimau

```



```

lb $t2, 0($s2) # $t2 lưu giá trị của từng phần tử trong string1
CheckD: bgt $t0, 22, CheckC #kiểm tra hết chuỗi D chưa
        beq $t2, $t5, fixD
        j Tmpdoimau
CheckC: bgt $t0, 42, CheckE #kiểm tra hết chuỗi C chưa
        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 chuỗi tiếp theo
          addi $t0, $t0, 1# biến đếm chuỗi
          j Inlhangdoimau
Enddoimau: li $v0, 4
          addi $a0 $s2 -64
          syscall
          addi $s0 $s0 1 # tăng biến đếm hàng lên 1
          j Lapdoimau
updatemau: move $t5 $s3
          move $t6 $s4
          move $t7 $s5
          j main
Exit:

```

4. Kết quả

Yêu cầu 1: Hiện thị hình ảnh chữ DCE trên giao diện console.

```

-----MENU-----
1. In chuỗi
2. In DCE chỉ có viền *
3. Đổi vị trí DCE -> ECD
4. Thay đổi màu chuỗi
5. Thoát
Nhập giá trị: 1

*****
*****
*222222222222222*
*22222*****22222*
*22222*      *22222*
*22222*      *22222*      *****
*22222*      *22222*      **11111*****111*
*22222*      *22222*      **1111**      **
*22222*      *22222*      *1111*
*22222*****22222*      *11111*
*2222222222222222*      *11111*
*****
*****
---
/ o o |
| > /
-----

*****
*333333333333333*
*33333*****
*33333*
*33333*****
*333333333333333*
*33333*****
*33333*
*33333*****
*333333333333333*
*****
*1111**
*1111****      *****
**11111***111*
*****
dce.hust.edu.vn

```

Yêu cầu 2: Sửa hình ảnh để chữ DCE chỉ còn phần viền và hiển thị

[illegible]

Yêu cầu 3: Hoán đổi vị trí DCE thành ECD và hiển thị.

```

-----MENU-----
1. In chu
2. In DCE chi co vien *
3. Doi vi tri DCE -> ECD
4. Thay doi mau chu
5. Thoat

Nhap gia tri: 3
*****
*33333333333333*                *****
*33333*****      *22222222222222*
*33333*            *22222*****222222*
*33333*****      *22222*          *22222*
*33333333333333*    ***** *22222*          *22222*
*33333*****        **11111*****111**22222*          *22222*
*33333*             **1111**         ** *22222*          *22222*
*33333*****        *1111*           *22222*          *222222*
*33333333333333*   *11111*           *22222*****222222*
*****             *11111*           *2222222222222222*
                        *11111*           *****
                          *1111**              ---
                            *1111****       ***** / o o |
                              **111111***111*     | > /
dce.hust.edu.vn               *****          -----

```

Yêu cầu 4: Nhập màu cho từng chữ D, C, E từ bàn phím và hiển thị DCE với màu mới

```
-----MENU-----
1. In chu
2. In DCE chi co vien *
3. Doi vi tri DCE -> ECD
4. Thay doi mau chu
5. Thoat
Nhap gia tri: 4
Nhap màu cho chu D(0->9): 6
Nhap màu cho chu C(0->9): 8
Nhap màu cho chu E(0->9): 9

*****
*****
*6666666666666666*
*66666*****666666*
*66666*      *66666*
*66666*      *66666*      *****
*66666*      *66666*      **88888*****888*
*66666*      *66666*      **88888**      **
*66666*      *666666*      *88888*
*66666*****666666*      *88888*
*666666666666666666*      *88888*
*****
---
/ o o |
| > /
-----

*****
*999999999999999*
*99999*****
*99999*
*99999*****
*999999999999999*
*99999*****
*99999*****
*99999*****
*999999999999999*
*99999*****
*999999999999999*
*****
*88888**
*88888****      *****
**8888888***888*
*****
dce.hust.edu.vn
```