

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



BÁO CÁO PROJECT CUỐI KỲ
MÔN HỌC : THỰC HÀNH KIẾN TRÚC MÁY TÍNH

Đề tài 2 : *Vẽ hình trên màn hình Bitmap*

Đề tài 9 : *Vẽ hình bằng ký tự ASCII*

Giáo viên hướng dẫn : Lê Bá Vui

Sinh viên thực hiện :

1. Nguyễn Thị Nguyệt – MSSV: 20194640

2. Lê Vương Khánh – MSSV: 20204989

Hà Nội, 21 tháng 7 năm 2022



MỤC LỤC

Chương 1 : Đề tài 2 : Vẽ hình trên màn hình Bitmap.....

I. Các bước xây dựng chương trình.....

B1. Thiết lập các địa chỉ và giá trị có trong bài toán.....

B2. Xây dựng các hàm thuật toán có trong chương trình.....

1. Hàm khởi tạo tọa độ đường tròn.....

2. Hàm delay.....

3. Hàm đặt lại màu và vẽ đường tròn.....

4. Hàm nhập dữ liệu từ bàn phím.....

5. Hàm di chuyển vị trí

6. Hàm kiểm tra vị trí.....

7. Hàm vẽ đường tròn.....

II. Mã nguồn của chương trình.....

III. Chạy mô phỏng bài toán.....

Chương 2 : Đề tài 9 : Vẽ hình bằng ký tự ASCII

I. Ý tưởng xây dựng chương trình.....

II. Mã nguồn.....

CHƯƠNG 1:

Đề tài 2 :Vẽ hình trên màn hình Bitmap

Sinh viên thực hiện : Nguyễn Thị Nguyệt

Đề bài : Viết chương trình vẽ một quả bóng hình tròn 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 pixel 1x1.
- 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), tăng tốc độ(Z), giảm tốc độ(X) trong bộ giả lập Keyboard and Display MMIO Simulator).
- Vị trí bóng ban đầu ở giữa màn hình.

Bài làm :

I. CÁC BƯỚC XÂY DỰNG CHƯƠNG TRÌNH

B1: Thiết lập các địa chỉ và giá trị có trong bài toán:

- màu sắc (màu đen của nền màn hình và màu vàng của đường tròn)
- Ký tự (A, D, S, W, Z, X, ENTER)
- Tọa độ x ban đầu của tâm đường tròn lưu vào thanh ghi s0 =256
- Tọa độ y ban đầu của tâm đường tròn lưu vào thanh ghi s1 =256
- R= 25 là bán kính đường tròn lưu vào thanh ghi s2
- Chiều rộng màn hình = 512 lưu vào thanh ghi s3
- Chiều dài màn hình = 512 lưu vào thanh ghi s4
- Thanh ghi s5 lưu địa chỉ màu vàng
- Khoảng cách giữa hai đường tròn lưu thanh ghi s6
- Giá trị dịch chuyển theo chiều ngang tại thời điểm hiện tại lưu ở thanh ghi t7
- Giá trị dịch chuyển theo chiều dọc tại thời điểm hiện tại lưu ở thanh ghi t8
- Thời gian delay =50 ms lưu ở thanh ghi t9

B2: Xây dựng các hàm thuật toán có trong chương trình

1. Hàm khởi tạo tọa độ đường tròn

khởi_tạo:

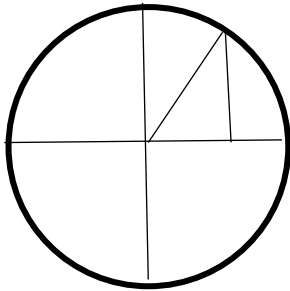
li	\$t0, 0	# khởi tạo i = 0
la	\$t5, Array	# lưu địa chỉ của mảng vào thanh ghi \$t5
loop:	# tạo vòng lặp chạy từ i đến R	
slt	\$v0, \$t0, \$s2	# v0=1 nếu i<R
beq	\$v0, \$zero, ket_thuc	# v0=0 <=> i >=R thì nhảy đến ket_thuc
mul	\$s6, \$s2, \$s2	# s6=R*R=R^2
mul	\$t3, \$t0, \$t0	# t3=i*i=i^2
sub	\$t3, \$s6, \$t3	# \$t3 = R^2 - i^2
move	\$v1, \$t3	# v1=t3
jal	sqrt	# nhảy đến hàm tính căn của t3
sw	\$a0, 0(\$t5)	# lấy giá trị của thanh ghi a0= sqrt(R^2 - i^2) lưu vào mảng dữ liệu
addi	\$t0, \$t0, 1	# i=i+1
add	\$t5, \$t5, 4	# đi đến vị trí tiếp theo của mảng dữ liệu
j	loop	
ket_thuc:		

Hàm tính căn của t3

sqrt:

```
mtc1    $v1, $f1          # đưa giá trị trong thanh ghi v0 vào thanh ghi f0
cvt.s.w  $f1, $f1          # chuyển giá trị của f0 tương đương với giá trị số nguyên 32 bit
sqrt.s   $f1, $f1          # Tính căn bậc hai của giá trị thanh ghi f0
cvt.w.s  $f1, $f1          # Chuyển f0 về dạng 32-bit
mfc1     $a0, $f1          # đưa giá trị thanh ghi a0=f0
jr       $ra
```

- Mục đích : xác định tọa độ các điểm của đường tròn và lưu vào bộ nhớ trong mảng array phục vụ việc vẽ đường tròn ở phần sau



Tìm vị trí của t3 = cách lấy $R^2 - i^2$

2. Hàm delay

```
.macro delay(%r)
    addi    $a0,%r,0
    li      $v0, 32
    syscall
.end_macro
```

- Mục đích : khi bóng dịch chuyển từ vị trí này qua vị trí khác thì chương trình sẽ dừng khoảng 50ms sau đó mới vẽ đường tròn ở vị trí mới
- Thời gian delay lưu ở thanh ghi %r

3. Hàm datmauveduongtron

```
.macro datmauveduongtron(%color)
    li      $s5, %color
    jal     ham_ve_duong_tron
.end_macro
```

- Mục đích : Khi bóng dịch chuyển đi, thì mình sẽ vẽ đường tròn cũ màu vàng thay bằng màu trùng với màu đen của màn hình, còn đường tròn mới màu vàng sẽ được vẽ ở vị trí khác
- Địa chỉ của màu lưu ở thanh ghi %color

4. Hàm nhập dữ liệu từ bàn phím

```
doc_ky_tu:
    lw      $k1, KEY_READY    # kiểm tra đã nhập kí tự nào chưa?
    beqz    $k1, check_vi_tri # Nếu k1!=0 => đã nhập ký tự thì nhảy đến hàm kiểm tra vị trí
    lw      $k0, KEY_CODE      # thanh ghi k0 lưu giá trị kí tự nhập vào
    beq     $k0, KEY_A, case_a  # đi chuyển qua trái
    beq     $k0, KEY_D, case_d  # đi chuyển qua phải
    beq     $k0, KEY_S, case_s  # đi chuyển xuống dưới
    beq     $k0, KEY_W, case_w  # đi chuyển lên trên
    beq     $k0, KEY_X, case_x  # Giảm tốc độ
    beq     $k0, KEY_Z, case_z  # Tăng tốc độ
    beq     $k0, KEY_ENTER, case_enter # Dừng chương trình
    j       check_vi_tri
    nop
case_a:
    jal     di_sang_trai
```

```

        j            check_vi_tri

case_d:
    jal            di_sang_phai
    j            check_vi_tri
case_s:
    jal            di_chuyen_xuong
    j            check_vi_tri
case_w:
    jal            di_chuyen_len
    j            check_vi_tri
case_x:
    addi           $t9,$t9,-10
    j            check_vi_tri
case_z:
    addi           $t9,$t9,20
    j            check_vi_tri
case_enter:
    j            endProgram
endProgram:
    li            $v0, 10
    syscall
➤ Mô tả : có 7 ký tự sẽ xuất hiện trong bài toán
    Ký tự chữ a : di chuyển qua bên trái
    Ký tự chữ d : di chuyển qua bên phải
    Ký tự chữ s : di chuyển xuống dưới
    Ký tự chữ w : di chuyển lên trên
    Ký tự chữ x : giảm tốc độ dịch chuyển của bóng
    Ký tự chữ z : Tăng tốc độ dịch chuyển của bóng
    Enter : kết thúc chương trình

```

5. Hàm di chuyển vị trí

```

di_sang_trai:
    sub           $t7, $zero, $t6
    li            $t8, 0
    jr            $ra
di_sang_phai:
    add           $t7, $zero, $t6
    li            $t8, 0
    jr            $ra
di_chuyen_len:
    li            $t7, 0
    sub           $t8, $zero, $t6
    jr            $ra
di_chuyen_xuong:
    li            $t7, 0
    add           $t8, $zero, $t6
    jr            $ra

```

6. Hàm kiểm tra vị trí

```

check_vi_tri:
phia_ben_phai:
    add           $v0, $s0, $s2          # v0=x0 + R , toa do tam hien tai+ ban kinh
    add           $v0, $v0,$t7          # neu x0 + R + khoang_cach > 512 thi nhay den ham di_sang_trai
    slt           $v1, $v0,$s3          # v1=1 neu v0< 512
    bne           $v1, $zero,phia_ben_trai

```

```

jal      di_sang_trai

nop
phia_ben_trai:
sub      $v0, $s0, $s2      # v0=x0-R
add      $v0, $v0, $t7      # neu x0 - R + khoang_cach < 0 thi nhay den ham di_sang_phai
slt      $v1, $v0, $zero     # v1=1 neu v0< 0
beq      $v1, $zero, phia_tren
jal      di_sang_phai
nop
phia_tren:
sub      $v0, $s1, $s2      # v0=y0 - R
add      $v0, $v0, $t8      # neu y0 - R + khoang_cach < 0 thi nhay den ham di_chuyen_len
slt      $v1, $v0, $zero     # v1=1 neu v0< 0
beq      $v1, $zero, phia_duoi
jal      di_chuyen_xuong
nop
phia_duoi:
add      $v0, $s1, $s2      # v0 = y0 + R
add      $v0, $v0, $t8      # neu y0 + R + khoang_cach > 512 thi nhay den ham
di_chuyen_xuong
slt      $v1, $v0,$s4      # v1=1 neu v0< 512
bne      $v1, $zero,draw
jal      di_chuyen_len
nop

```

- Mục đích kiểm tra lần lượt từng cạnh theo chiều dọc và chiều ngang màn hình . Ví dụ :
 Khi quả bóng đang dịch chuyển sang phải, Nếu tọa độ hiện tâm hiện tại + bán kính+ khoảng cách mà lớn hơn 512 thì quả bóng phải bật lại về theo hướng ngược lại , ở thuật toán di_sang_trai sẽ làm điều này
 Còn nếu chưa lớn hơn 512 thì lần lượt xuống kiểm tra các cạnh còn lại .

7. Hàm vẽ đường tròn

```

draw:
datmauveduongtron(BACKGROUND) # ve duong tron trung mau nen
add      $s0, $s0, $t7      # Cap nhat toa do x cua duong tron
add      $s1, $s1, $t8      # cap nhat toa do y cua duong tron

datmauveduongtron(YELLOW)    # ve duong tron moi mau vang
delay($t9)                    # dung 1 khoang thoi gian roi ve duong tron moi
j Start

```

Gọi 2 hàm khởi tạo lúc ban đầu để vẽ đường tròn cũ trùng với màu nền và đường tròn mới có màu vàng ở vị trí mới.

```

ham_ve_duong_tron:
add      $sp, $sp, -4
sw       $ra, 0($sp)
li       $t0, 0              # khoi tao bien i = 0

loop_ve_duong_tron:
slt      $v0, $t0, $s2      # v0=1 neu i< R
beq      $v0, $zero, ket_thuc_ve# neu v0=0 <=> i>=R => ket_thuc_ve
sll      $t5, $t0, 2
lw       $t3, Array($t5)    # nap sqrt(R^2-i^2) luu o Array vao thanh ghi $t3
move     $a0, $t0            # i = #t0= $a0
move     $a1, $t3            # j = $t3= $a1
jal      ve_diem             # ve 2 diem (x0 + i, y0 + j), (x0 + j, y0 + i) tren phan tu thu I
sub      $a1, $zero, $t3

```

```

jal    ve_diem                # ve 2 diem (x0 + i, y0 - j), (x0 + j, y0 - i) tren phan tu thu II

sub    $a0, $zero, $t0
jal    ve_diem                # ve 2 diem (x0 - i, y0 - j), (x0 - j, y0 - i) tren phan tu thu III
add    $a1, $zero, $t3
jal    ve_diem                # ve 2 diem (x0 - i, y0 + j), (x0 - j, y0 + i) tren phan tu thu IV
addi   $t0, $t0, 1
j      loop_ve_duong_tron

ket_thuc_ve:
lw     $ra, 0($sp)
add    $sp, $sp, 0
jr     $ra

```

#hàm vẽ điểm của đường tròn

ve_diem:

```

add    $t1, $s0, $a0        # xi = x0 + i
add    $t4, $s1, $a1        # yi = y0 + j
mul    $t2, $t4, $s3        # yi * SCREEN_WIDTH
add    $t1, $t1, $t2        # yi * SCREEN_WIDTH + xi (Toa do 1 chieu cua diem anh)
sll    $t1, $t1, 2          # dia chi tuong doi cua diem anh
sw     $s5, SCREEN($t1)     # ve anh
add    $t1, $s0, $a1        # xi = x0 + j
add    $t4, $s1, $a0        # yi = y0 + i
mul    $t2, $t4, $s3        # yi * SCREEN_WIDTH
add    $t1, $t1, $t2        # yi * SCREEN_WIDTH + xi (Toa do 1 chieu cua diem anh)
sll    $t1, $t1, 2          # dia chi tuong doi cua diem anh
sw     $s5, SCREEN($t1)     # ve anh
jr     $ra

```

- Sẽ chia đường tròn thành 4 phần , phần tư thứ I,II,III,IV và áp dụng thuật toán tìm tọa độ điểm để vẽ được toàn bộ đường tròn

II. MÃ NGUỒN CỦA CHƯƠNG TRÌNH

```

.eqv SCREEN 0x10010000
.eqv YELLOW 0x00FFFF66
.eqv BACKGROUND 0x00000000
# Thiet lap ky tu
.eqv KEY_A 0x00000061      # di chuyen sang trai
.eqv KEY_D 0x00000064      # di chuyen sang phai
.eqv KEY_S 0x00000073      # di chuyen xuong duoi
.eqv KEY_W 0x00000077      # di chuyen len tren
.eqv KEY_Z 0x0000007A      # tang toc do di chuyen
.eqv KEY_X 0x00000078      # giam toc do di chuyen
.eqv KEY_ENTER 0x0000000A   # chuong trinh dung lai
# thiet lap khoang cach giua hai duong tron
.eqv khoang_cach 20
.eqv KEY_CODE 0xFFFF0004
.eqv KEY_READY 0xFFFF0000
#=====
.data
    Array: .space 512      #cap bo nho luu toa do cac diem cua duong tron
.text
    li     $s0, 256        # x = 256 khoi tao toa do x ban dau cua tam duong tron
    li     $s1, 256        # y = 256 khoi tao toa do y ban dau cua tam duong tron

```

```

li    $s2, 25      # R = 25 R la ban kinh cua duong tron
li    $s3, 512     # SCREEN_WIDTH = 512 chieu rong man hinh
li    $s4, 512     # SCREEN_HEIGHT = 512 chieu dai man hinh
li    $s5, YELLOW  # duong tron co mau vang
li    $t6, khoang_cach # Khoang cach giua cac hinh tron
li    $t7, 0       # dx = 0 gia tri dich chuyen theo chieu ngang tai thoi diem hien tai
li    $t8, 0       # dy = 0 gia tri dich chuyen theo chieu doc tai thoi diem hien tai
li    $t9, 50      # Thanh ghi luu tru thoi gian delay

```

```

#=====
# HAM KHOI TAO TOA DO DUONG TRON
#=====

```

```

khoi_tao:
li    $t0, 0       # khoi tao i = 0
la    $t5, Array   # luu dia chi cua mang vao thanh ghi $t5
loop: # tao vong lap chay tu i den R
slt   $v0, $t0, $s2 # v0=1 neu i<R
beq   $v0, $zero, ket_thuc # v0=0 <=> i >= R thi nhay den ket_thuc
mul   $s6, $s2, $s2 # s6=R*R=R^2
mul   $t3, $t0, $t0 # t3=i*i=i^2
sub   $t3, $s6, $t3 # $t3 = R^2 - i^2
move  $v1, $t3     # v1=t3
jal   sqrt         # nhay den ham tinh can cua t3

sw    $a0, 0($t5)  # lay gia tri cua thanh ghi a0= sqrt(R^2 - i^2) luu vao mang du lieu
addi  $t0, $t0, 1  # i=i+1
add   $t5, $t5, 4  # di den vi tri tiep theo cua mang du lieu
j     loop

```

```
ket_thuc:
```

```

#-----
# tao ham lam cho chuong trinh dung chay trong 1 khoang thoi gian
# thoi gian co gia tri luu o thanh ghi %r khi goi ham

```

```

.macro delay(%r)
addi  $a0, %r, 0
li    $v0, 32
syscall

```

```
.end_macro
```

```
#tao ham de dat lai mau va ve them duong tron o vi tri moi
```

```
#dia chi cua mau luu o thanh ghi %color khi goi ham
```

```

.macro datmauveduongtron(%color)
li    $s5, %color
jal   ham_ve_duong_tron

```

```
.end_macro
```

```

#=====
# HAM NHAP DU LIEU TU BAN PHIM
#=====

```

```
Start :
```

```

doc_ky_tu:
lw    $k1, KEY_READY # kiem tra da nhap ki tu nao chua?
beqz  $k1, check_vi_tri # Neu k1!=0 =>da nhap ky tu thi nhay den ham kiem tra vi tri
lw    $k0, KEY_CODE   # thanh ghi k0 luu gia tri ki tu nhap vao
beq   $k0, KEY_A, case_a # di chuyen qua trai
beq   $k0, KEY_D, case_d # di chuyen qua phai

```



```

        beq    $k0, KEY_S, case_s    # di chuyen xuong duoi
        beq    $k0, KEY_W, case_w    # di chuyen len tren
        beq    $k0, KEY_X, case_x    # Giam toc do
        beq    $k0, KEY_Z, case_z    # Tang toc do
        beq    $k0, KEY_ENTER, case_enter # Dung chuong trinh
        j      check_vi_tri
        nop
case_a:
        jal    di_sang_trai
        j      check_vi_tri
case_d:
        jal    di_sang_phai
        j      check_vi_tri
case_s:
        jal    di_chuyen_xuong
        j      check_vi_tri
case_w:
        jal    di_chuyen_len
        j      check_vi_tri
case_x:
        addi   $t9,$t9,-10
        j      check_vi_tri
case_z:
        addi   $t9,$t9,20
        j      check_vi_tri
case_enter:
        j      endProgram
endProgram:
        li     $v0, 10
        syscall

```

```

#=====

```

```

# HAM KIEM TRA VI TRI

```

```

#=====

```

```

check_vi_tri:
phia_ben_phai:
        add    $v0, $s0, $s2        # v0=x0 + R , toa do tam hien tai+ ban kinh
        add    $v0, $v0,$t7        # neu x0 + R + khoang_cach > 512 thi nhay den ham di_sang_trai
        slt    $v1, $v0,$s3        # v1=1 neu v0< 512
        bne    $v1, $zero,phia_ben_trai
        jal    di_sang_trai
        nop
phia_ben_trai:
        sub    $v0, $s0, $s2        # v0=x0-R
        add    $v0, $v0, $t7        # neu x0 - R + khoang_cach < 0 thi nhay den ham di_sang_phai
        slt    $v1, $v0, $zero      # v1=1 neu v0< 0
        beq    $v1, $zero, phia_tren
        jal    di_sang_phai
        nop
phia_tren:
        sub    $v0, $s1, $s2        # v0=y0 - R
        add    $v0, $v0, $t8        # neu y0 - R + khoang_cach < 0 thi nhay den ham di chuyen len
        slt    $v1, $v0, $zero      # v1=1 neu v0< 0
        beq    $v1, $zero, phia_duoi

```

```

        jal      di_chuyen_xuong
        nop
phia_duoi:
        add      $v0, $s1, $s2      # v0 = y0 + R
        add      $v0, $v0, $t8      # neu y0 + R + khoang_cach > 512 thi nhay den ham di_chuyen_xuong
        slt      $v1, $v0, $s4      # v1=1 neu v0< 512
        bne      $v1, $zero, draw
        jal      di_chuyen_len
        nop

```

```

#=====
# HAM VE DUONG TRON
#=====

```

```

draw:
        datmauveduongtron(BACKGROUND) # ve duong tron trung mau nen
        add      $s0, $s0, $t7      # Cap nhat toa do x cua duong tron
        add      $s1, $s1, $t8      # cap nhat toa do y cua duong tron

        datmauveduongtron(YELLOW)   # ve duong tron moi mau vang
        delay($t9)                   # dung 1 khoang thoi gian roi ve duong tron moi
        j Start

```

```

ham_ve_duong_tron:
        add      $sp, $sp, -4
        sw      $ra, 0($sp)
        li      $t0, 0              # khoi tao bien i = 0
loop_ve_duong_tron:
        slt      $v0, $t0, $s2      # v0=1 neu i< R
        beq      $v0, $zero, ket_thuc_ve # neu v0=0 <=> i>=R => ket_thuc_ve
        sll      $t5, $t0, 2
        lw      $t3, Array($t5)      # nap sqrt(R^2-i^2) luu o Array vao thanh ghi $t3
        move     $a0, $t0              # i = #t0= $a0
        move     $a1, $t3              # j = $t3= $a1
        jal      ve_diem              # ve 2 diem (x0 + i, y0 + j), (x0 + j, y0 + i) tren phan tu thu I
        sub      $a1, $zero, $t3
        jal      ve_diem              # ve 2 diem (x0 + i, y0 - j), (x0 + j, y0 - i) tren phan tu thu II
        sub      $a0, $zero, $t0
        jal      ve_diem              # ve 2 diem (x0 - i, y0 - j), (x0 - j, y0 - i) tren phan tu thu III
        add      $a1, $zero, $t3
        jal      ve_diem              # ve 2 diem (x0 - i, y0 + j), (x0 - j, y0 + i) tren phan tu thu IV
        addi     $t0, $t0, 1
        j        loop_ve_duong_tron
ket_thuc_ve:
        lw      $ra, 0($sp)
        add     $sp, $sp, 0
        jr      $ra

```

```

# Ham ve diem tren duong tron
ve_diem:

```

```

        add      $t1, $s0, $a0      # xi = x0 + i
        add      $t4, $s1, $a1      # yi = y0 + j
        mul      $t2, $t4, $s3      # yi * SCREEN_WIDTH
        add      $t1, $t1, $t2      # yi * SCREEN_WIDTH + xi (Toa do 1 chieu cua diem anh)

```

```

sll    $t1, $t1, 2          # dia chi tuong doi cua diem anh
sw     $s5, SCREEN($t1)    # ve anh
add    $t1, $s0, $a1        # xi = x0 + j
add    $t4, $s1, $a0        # yi = y0 + i
mul    $t2, $t4, $s3        # yi * SCREEN_WIDTH
add    $t1, $t1, $t2        # yi * SCREEN_WIDTH + xi (Toa do 1 chieu cua diem anh)
sll    $t1, $t1, 2          # dia chi tuong doi cua diem anh
sw     $s5, SCREEN($t1)    # ve anh
jr     $ra

```

#-----

Ham tinh can cua t3

sqrt:

```

mtc1   $v1, $f1            # dua gia tri trong thanh ghi v0 vao thanh ghi f0
cvt.s.w $f1, $f1           # chuyen gia tri cua f0 tuong duong voi gia tri so nguyen 32 bit
sqrt.s  $f1, $f1           # Tinh can bac hai cua gia tri thanh ghi f0
cvt.w.s $f1, $f1           # Chuyen f0 ve dang 32-bit
mfc1    $a0, $f1           # dat gia tri thanh ghi a0=f0
jr      $ra

```

#=====

#CAC HAM DI CHUYEN

#=====

di_sang_trai:

```

sub     $t7, $zero, $t6
li      $t8, 0
jr      $ra

```

di_sang_phai:

```

add     $t7, $zero, $t6
li      $t8, 0
jr      $ra

```

di_chuyen_len:

```

li      $t7, 0
sub     $t8, $zero, $t6
jr      $ra

```

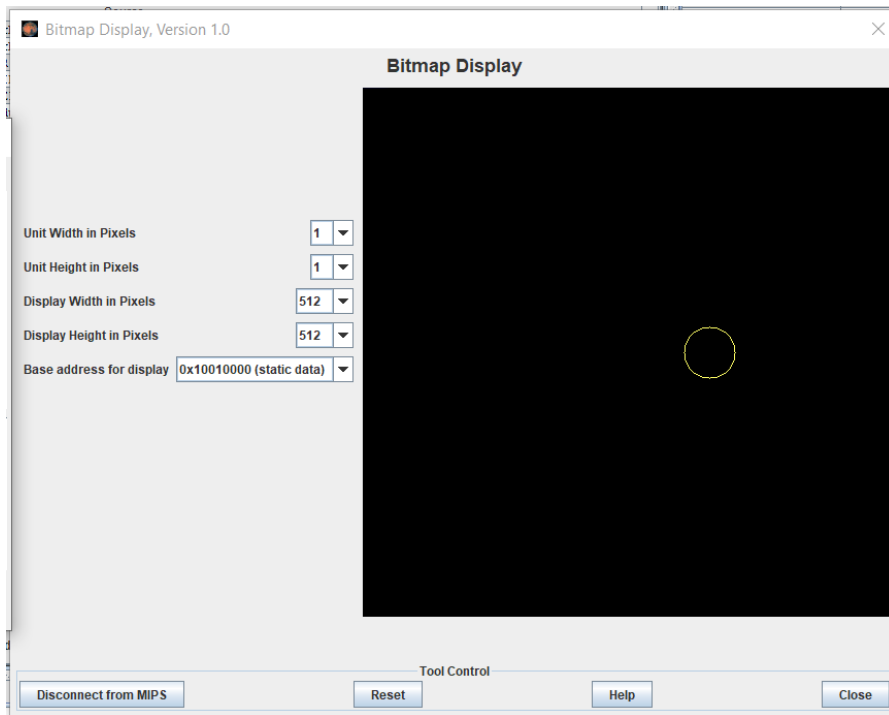
di_chuyen_xuong:

```

li      $t7, 0
add     $t8, $zero, $t6
jr      $ra

```

III. CHẠY MÔ PHỎNG BÀI TOÁN



CHƯƠNG 2 :

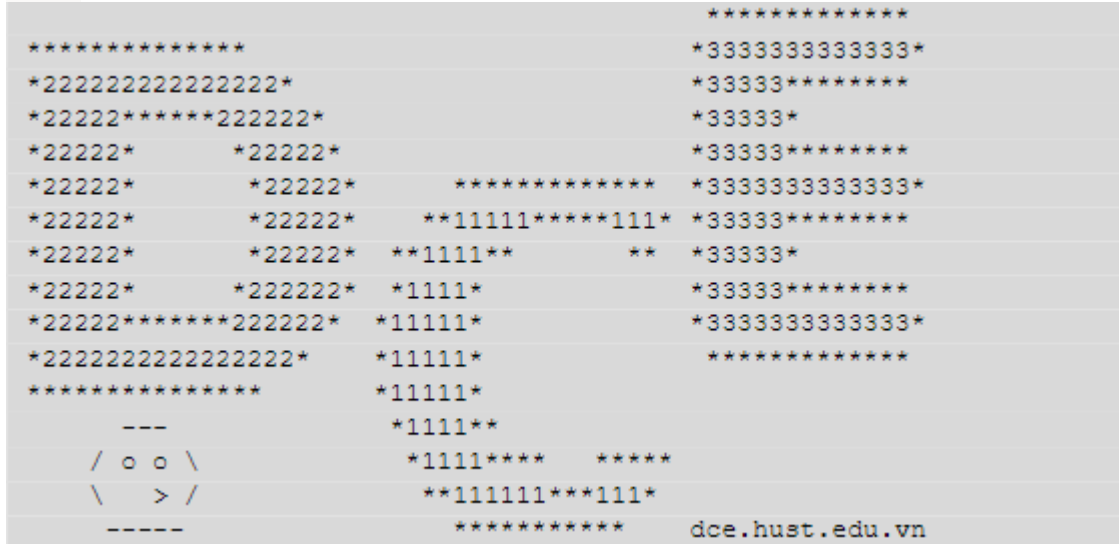
Đề tài 9 : Vẽ hình bằng ký tự ASCII

Sinh viên thực hiện : Lê Vương Khánh

Mssv: 20204989

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



- 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 họa 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 nhỏ hơn chứa ảnh được chứa sẵn trong code, không được tạo thêm vùng nhỏ mới để chứa ảnh hiệu chỉnh.

I. Ý TƯỞNG

Câu 1: Ta sẽ nhập dữ liệu giống như đề bài vào mã nguồn mips, sau đó thực hiện in ra trên màn hình

```
.data
line1: .asciiz "*****\n"
line2: .asciiz "3333333333333\n"
line3: .asciiz "2222222222222\n"
line4: .asciiz "22222*****22222\n"
line5: .asciiz "22222*      *22222*      33333*****\n"
line6: .asciiz "22222*      *22222*      ***** 333333333333*\n"
line7: .asciiz "22222*      *22222*      **11111*****111* 33333*****\n"
line8: .asciiz "22222*      *22222*      **1111**      ** 33333*\n"
line9: .asciiz "22222*      *22222*      *1111*      33333*****\n"
line10: .asciiz "22222*****22222*      *11111*      333333333333*\n"
line11: .asciiz "2222222222222222*      *11111*      *****\n"
line12: .asciiz "*****\n"
line13: .asciiz "      *1111*\n"
line14: .asciiz "      *1111****      *****\n"
line15: .asciiz "      **111111***111*\n"
line16: .asciiz "*****      dce.hust.edu.vn\n"
menu: .asciiz "menu: \n1. Chỉ con lai vien\n2. Doi vi tri thanh ECD\n3. Doi ki tu mau\n4. reset\n5. thoat\nChon chuc nang (1-5) : "
nhap1: .asciiz "Mau chu D(0-9): "
nhap2: .asciiz "Mau chu C(0-9): "
nhap3: .asciiz "Mau chu E(0-9): "
nhap_sai: .asciiz "Ky tu ban nhap khong hop le, nhap lai: \n"
thong_bao: .asciiz "==== Khong co chuc nang nay =====\n"
```

Câu 2: thực hiện chức năng hiển thị chữ to (chỉ duyệt viền):

Ý tưởng: ban đầu khởi tạo các chữ DCE theo câu 1 (hình bên trên)

Sau đó tạo ra function duyệt, đối tượng được duyệt là các chữ số từ 0-9 nằm trong các dòng thuộc mảng khởi tạo DCE theo câu 1.

Nếu duyệt được bất kì chữ số nào từ 0-9 nằm trong khoảng trên, ta sẽ thay đổi từ chữ số đó sang kí tự khoảng trắng

```

        j      $ra0,1
TIE:      # xu li vien line i (la $a1, line i)
        li $t0,0
        la $a1,line1
        jal XU_LY_VIEN
        la $a1,line2
        jal XU_LY_VIEN
        la $a1,line3
        jal XU_LY_VIEN
        la $a1,line4
        jal XU_LY_VIEN
        la $a1,line5
        jal XU_LY_VIEN
        la $a1,line6
        jal XU_LY_VIEN
        la $a1,line7
        jal XU_LY_VIEN
        la $a1,line8
        jal XU_LY_VIEN
        la $a1,line9
        jal XU_LY_VIEN
        la $a1,line10
        jal XU_LY_VIEN
        la $a1,line11
        jal XU_LY_VIEN
        la $a1,line12
        jal XU_LY_VIEN
        la $a1,line13
        jal XU_LY_VIEN
        la $a1,line14
        jal XU_LY_VIEN
        la $a1,line15

```

```

# i = 0
# xu li vien line1
# xu li vien line2
# xu li vien line3
#...

```

Xử lý với từng dòng

```

XU_LY_VIEN:      add $t1,$a1,$t0
                  lb $a0,0($t1)
                  li $t1,10
                  beq $a0,$t1,end_xu_ly_vien
                  li $t1,48
                  slt $t2,$a0,$t1
                  li $t1,57
                  slt $t3,$t1,$a0
                  or $t1,$t2,$t3
                  beq $t1,$0,print_32
                  li $v0,11
                  syscall
                  j jump_xu_ly_vien
print_32:         li $v0,11
                  li $a0,32
                  syscall
jump_xu_ly_vien: addi $t0,$t0,1
                  j XU_LY_VIEN
end_xu_ly_vien:  li $v0,11
                  syscall

```

```

# thuc hien duyet tung ki tu - $a1 la bien luu tru line i
# t1 = '\n'
# if($t1 = "\n") end_xy_ly vien dong i
# $t2 = $a0 < 48 ? 1:0
# $t3 = 57 < $a0 ? 1:0
# if(48<= $a0 <= 57) print " "
# if($a0<48 || $a0 >57) print $a0
# quay lai vong lap
# print " "
# quay lai vong lap
# print "\n"

```

Hàm duyệt đổi từ chữ số chuyển sang khoảng trắng

Câu 3: để thực hiện chức năng đảo, ta dùng phép biến đổi tịnh tiến

Cách tiến hành: tạo ra hàm swap:

Với chữ D: do D có kí tự tối đa là 22, nên ta sẽ thực hiện đảo từ ô nhớ 0 đến 22 trong dòng, sau đó định tiến lên 43 đơn vị. Ví dụ line1[0] = line1[43] và line1[43] = line1[0].

Tương tự với các chữ C và E

```

huc_nang_1:    j    VIEN                                # jump vien
WAP:           # thuc hien dao tung long (line i)
               la    $a0,line1
               jal    DAO                                # swap dong thu 1
               la    $a0,line2
               jal    DAO                                #swap dong thu 2
               la    $a0,line3
               jal    DAO                                #swap dong thuc 3
               la    $a0,line4
               jal    DAO                                #...
               la    $a0,line5
               jal    DAO
               la    $a0,line6
               jal    DAO
               la    $a0,line7
               jal    DAO
               la    $a0,line8
               jal    DAO
               la    $a0,line9
               jal    DAO
               la    $a0,line10
               jal    DAO
               la    $a0,line11
               jal    DAO
               la    $a0,line12
               jal    DAO
               la    $a0,line13
               jal    DAO
               la    $a0,line14
               jal    DAO
               la    $a0,line15
               jal    DAO
               la    $a0,line16
               jal    DAO
               li    $t1,3

```

ứng dụng hàm Dao với từng dòng

```

00  DAO:    li    $t0,0                                # khoi tao i = 0
01  loop2:  add    $t1,$a0,$t0                          # duyet tung phan tu
02          #dinh tien cac ki tu len 43 o nho va ca ki tu tu 43 tro len di chuyen xuong 43 o nho
03          lb    $t2,0($t1)
04          lb    $t3,43($t1)                            # doi vi tri 2 ki tu cach nhau 43 o nho
05          sb    $t3,0($t1)
06          sb    $t2,43($t1)
07          li    $t1,21                                # thuc hien voi 21 ki tu (do ki tu lon nhat can de
08          beq    $t0,$t1,end_loop2
09          addi   $t0,$t0,1                            # i = i + 1
10          j      loop2
11  end_loop2: jr    $ra                                # quay lai duyet dong tiep theo
-----

```

Hàm Dao: thực hiện định tiến các ô nhớ

Câu 4: thực hiện thay màu cho các chữ cái:

Ban đầu cho người dùng nhập màu từ 0-9, nếu nhập sai yêu cầu người dùng nhập lại

Với chữ D, các kí tự được thay thế sẽ là từ 0 đến 22(như ý 2)

Với chữ C, các kí tự được thay thế sẽ là từ 22 đến 41

```

        j loop
nhap_mau:    li $v0,4
             syscall
             li $v0,12
             syscall
             add $a2,$v0,$0
             li $v0,4
             la $a0, xuong_dong
             syscall
kiem_tra:    li $t1,48
             slt $t7,$a2,$t1
             li $t1,57
             slt $t6, $t1, $a2
             or  $t7,$t7,$t6
             bne $t7,$0,nhap_lai
             jr  $ra
end_nhap:
nhap_lai:    li $v0,4
             la $a0,nhap_sai
             syscall
             add $a0,$a1,$0
             j  nhap_mau
MAU1:       add $v1, $s3,$0
             j  XET1
MAU2:       add $v1, $s4,$0
             j  XET2
MAU3:       add $v1, $s5,$0
             j  XET3
XET1:       li $t0,0
             li $a2,0
             li $a3,22
             add $t7,$zero,$s0
             jal XU_LY
             add $s0,$v1,$0
             li $t1,0
             bne $t1,$0,nhap_lai
             jr  $ra

XU_LY:
             add $k1,$0,$ra
             la $k0, line1
             jal duyet
             la $k0, line2
             jal duyet
             la $k0, line3
             jal duyet
             la $k0, line4
             jal duyet
             la $k0, line5
             jal duyet
             la $k0, line6
             jal duyet
             la $k0, line7
             jal duyet
             la $k0, line8
             jal duyet
             la $k0, line9
             jal duyet
             la $k0, line10
             jal duyet
             la $k0, line11
             jal duyet
             la $k0, line12
             jal duyet
             la $k0, line13
             jal duyet
             la $k0, line14
             jal duyet
             la $k0, line15
             jr  $ra

# print "mau chu D(0-9): "
# nhap ma mau
# print "xuong_dong"
# $t1 = '0'
# $t7 = $v0 < 0 ? 1 : 0
# $t6 = 9 < $v0 ? 1 : 0
# $t7 = $t7 and $t6
# if($v0>9 || $v0 < 0) nhap_lai
# print thong bao nhap sai
# $v1 la bien thay doi mau
# thuc hien thay doi mau cho chu D
# i = 0
# xet tu ki tu 0 - 22 (so ky tu cua chu D)
# xet tu ki tu 0 - 22
# $t7 la bien luu mau cua cac chu
# bat dau doi mau
# thuc hien duyet tung dong
# luu thanh ghi $ra de quay lai (XET i)

```

Hàm duyệt từng dòng(giống ý 2)


```

249      add $ra,$0,$k1
250      jr $ra                                # tiếp tục xét màu của kí tự tiếp theo
251  duyet: add $t1,$k0,$t0                    # address i # duyệt từng kí tự trong khoảng $a2 - $a3
252      lb $t2,0($t1)                        # value i
253      beq $t0,$a3, end_duyet              # đến $a3 dừng lại
254      beq $t2,$t7,change                  # thay đổi màu
255  jump: addi $t0,$t0,1
256      j duyet
257  change: sb $v1, 0($t1)

```

Hàm thay đổi chữ số

II. MÃ NGUỒN

.data

```

line1: .asciiz "*****\n"
line2: .asciiz "***** *333333333333* \n"
line3: .asciiz "*22222222222222* *33333***** \n"
line4: .asciiz "*22222*****22222* *33333* \n"
line5: .asciiz "*22222* *22222* *33333***** \n"
line6: .asciiz "*22222* *22222* ***** *333333333333* \n"
line7: .asciiz "*22222* *22222* **11111*****111* *33333***** \n"
line8: .asciiz "*22222* *22222* **1111** ** *33333* \n"
line9: .asciiz "*22222* *222222* *1111* *33333***** \n"
line10: .asciiz "*22222*****222222* *11111* *33333333333333* \n"
line11: .asciiz "*222222222222222222* *11111* ***** \n"
line12: .asciiz "***** *11111* \n"
line13: .asciiz " --- *1111** \n"
line14: .asciiz " / o o \\\n"
line15: .asciiz " \\\n"
line16: .asciiz " ---- ***** dce.hust.edu.vn \n"

```

menu: .asciiz "menu: \n1. Chi con lai vien\n2. Doi vi tri thanh ECD\n3. Doi ki tu mau\n4. reset\n5. thoat\nChon chuc nang (1-5) : "

nhap1: .asciiz "Mau chu D(0-9): "

nhap2: .asciiz "Mau chu C(0-9): "

nhap3: .asciiz "Mau chu E(0-9): "

nhap_sai: .asciiz "Ky tu ban nhap khong hop le, nhap lai: \n"

thong_bao: .ascii "===== Không có chức năng này =====\n"

xuong_dong: .ascii "\n"

.text

li \$s0,50 # mau D = 2

li \$s1,49 # mau C = 1

li \$s2,51 # mau E = 3

li \$t0,0 # i = 0

li \$s6,0 # kiểm tra chức năng 2 có bat hay không

li \$a3,21

j printf

loop:

li \$v0,4

la \$a0,menu # hiện menu

syscall

li \$v0,5 # chọn chức năng

syscall

add \$s7,\$v0,\$0 # lưu chức năng

li \$t1,1

beq \$v0,\$t1,chuc_nang_1 # nhận 1 thực hiện chức năng 1

li \$t1,2

bne \$v0,\$t1,END_ADD # nếu không phải chức năng 2 thì bỏ qua
ADD

ADD: add \$v0,\$v0,\$s6 # nếu là chức năng 2 thì ADD \$v0 = \$v0 + \$s6

END_ADD:

li \$t1,2

beq \$v0,\$t1,SWAP # nếu \$v0 là chức năng 2 và \$s6 = 0 (chưa
SWAP) thì thực hiện SWAP

li \$t1,7

beq \$v0,\$t1,chuc_nang_2	# neu \$v0 = 7 (\$s6 = 5) da SWAP roi nen chi can
print	
li \$t1,3	
beq \$v0,\$t1,chuc_nang_3	# thuc hien chuc nang 3
li \$t1,4	
beq \$v0,\$t1,chuc_nang_4	# thuc hien chuc nang 4
li \$t1,5	
beq \$v0,\$t1,chuc_nang_5	# thuc hien chuc nang 5
j khong_co_chuc_nang	
chuc_nang_1: j VIEN	# jump vien
SWAP:	# thuc hien dao tung long (line i)
la \$a0,line1	
jal DAO	# swap dong thu 1
la \$a0,line2	
jal DAO	#swap dong thu 2
la \$a0,line3	
jal DAO	#swap dong thuc 3
la \$a0,line4	
jal DAO	#...
la \$a0,line5	
jal DAO	
la \$a0,line6	
jal DAO	
la \$a0,line7	
jal DAO	
la \$a0,line8	
jal DAO	
la \$a0,line9	
jal DAO	

```

        la $a0,line10

    jal DAO

    la $a0,line11

    jal DAO

    la $a0,line12

    jal DAO

    la $a0,line13

    jal DAO

    la $a0,line14

    jal DAO

    la $a0,line15

    jal DAO

    la $a0,line16

    jal DAO

    li $t1,3

    beq $s7,$t1,end_xu_ly_dao           # neu la chuc nang 3 thi dao lai lan nua

    li $t1,4

    beq $s7,$t1,reset                 # neu la chuc nang 4 thi dao ve vi tri ban dau

chuc_nang_2:  li $s6,5

                li $t7,7

                j printf                # print sau khi swap

chuc_nang_3:  bne $s6,$0,SWAP           # neu $s6 = 5 thi thuc hien dao ve vi tri ban dau
khoi tao

                j end_xu_ly_dao         # thuc hien nhap

xu_ly_dao:    li $s7,10

                j SWAP                  # sau khi dao ve vi tri ban dau thuc hien
swap

end_xu_ly_dao: la $a0,nhap1             # print "mau chu D"

                la $a1,nhap1           # bien phu luu lai de neu nhap loi co the print lai
dung mau

```

```

        jal nhap_mau                                # thuc hien nhap
        add $s3,$a2,$0                              # $s3 luu ma mau vua nhap
        la $a0,nhap2                                # tiep tục nhập màu các chu tiếp theo
        la $a1,nhap2
        jal nhap_mau
        add $s4,$a2,$0
        la $a0,nhap3
        la $a1,nhap3
        jal nhap_mau
        add $s5,$a2,$0
        j MAU1                                       # thuc hien thay doi mau

chuc_nang_4: bne $s6, $0,SWAP
        j reset

chuc_nang_5: li $v0,10
        syscall

reset:     li $s6,0
        li $s3,50                                # mau D = 2
        li $s4,49                                # mau C = 1
        li $s5,51                                # mau E = 3
        j MAU1

VIEN: # xu li vien line i (la $a1, line i)
        li $t0,0                                  # i = 0
        la $a1,line1
        jal XU_LY_VIEN                            # xu li vien line1
        la $a1,line2
        jal XU_LY_VIEN                            # xu li vien line2
        la $a1,line3
        jal XU_LY_VIEN                            # xu li vien line3

```

la \$a1,line4

jal XU_LY_VIEN

#...

la \$a1,line5

jal XU_LY_VIEN

la \$a1,line6

jal XU_LY_VIEN

la \$a1,line7

jal XU_LY_VIEN

la \$a1,line8

jal XU_LY_VIEN

la \$a1,line9

jal XU_LY_VIEN

la \$a1,line10

jal XU_LY_VIEN

la \$a1,line11

jal XU_LY_VIEN

la \$a1,line12

jal XU_LY_VIEN

la \$a1,line13

jal XU_LY_VIEN

la \$a1,line14

jal XU_LY_VIEN

la \$a1,line15

jal XU_LY_VIEN

la \$a1,line16

jal XU_LY_VIEN

j loop

nhap_mau: li \$v0,4

	syscall	# print "mau chu D(0-9): "
	li \$v0,12	
	syscall	# nhap ma mau
	add \$a2,\$v0,\$0	
	li \$v0,4	
	la \$a0, xuong_dong	# print "xuong_dong"
	syscall	
kiem_tra:	li \$t1,48	# \$t1 = '0'
	slt \$t7,\$a2,\$t1	# \$t7 = \$v0 < 0 ? 1 : 0
	li \$t1,57	
	slt \$t6, \$t1, \$a2	# t6 = 9 < \$v0 ? 1 : 0
	or \$t7,\$t7,\$t6	# \$t7 = \$t7 and \$t6
	bne \$t7,\$0,nhap_lai	# if(\$v0>9 \$v0 < 0) nhap_lai
	jr \$ra	
end_nhap:		
nhap_lai:	li \$v0,4	
	la \$a0,nhap_sai	# print thong bao nhap sai
	syscall	
	add \$a0,\$a1,\$0	
	j nhap_mau	
MAU1:	add \$v1, \$s3,\$0	# \$v1 la bien thay doi mau
	j XET1	# thuc hien thay doi mau cho chu D
MAU2:	add \$v1, \$s4,\$0	
	j XET2	
MAU3:	add \$v1, \$s5,\$0	
	j XET3	
XET1:	li \$t0,0	# i = 0
	li \$a2,0	# xet tu ki tu 0 - 22 (so ky tu cua chu D)

li \$a3,22

xet tu ki tu 0 - 22

add \$t7,\$zero,\$s0

\$t7 la bien luu mau cua cac chu

jal XU_LY

bat dau doi mau

add \$s0,\$v1,\$0

li \$t1,3

beq \$s7,\$t1,MAU2

li \$t1,4

beq \$s7,\$t1,MAU2

XET2: li \$t0,22

li \$a2,22

xet ki tu tu 22-41 (ki tu cua chu C)

li \$a3,41

add \$t7,\$zero,\$s1

jal XU_LY

add \$s1,\$v1,\$0

li \$t1,3

beq \$s7,\$t1,MAU3

li \$t1,4

beq \$s7,\$t1,MAU3

XET3: li \$t0,41

li \$a2,41

li \$a3,70

add \$t7,\$zero,\$s2

jal XU_LY

add \$s2,\$v1,\$0

bne \$s6,\$0,xu_ly_dao

neu \$s6 = 5 thuc hien dao lai vi tri

j printf

print

XU_LY:

thuc hien duyettung dong

add \$k1,\$0,\$ra

luu thanh ghi \$ra de quay lai (XET i)

la \$k0, line1

jal duyet

la \$k0, line2

jal duyet

la \$k0, line3

jal duyet

la \$k0, line4

jal duyet

la \$k0, line5

jal duyet

la \$k0, line6

jal duyet

la \$k0, line7

jal duyet

la \$k0, line8

jal duyet

la \$k0, line9

jal duyet

la \$k0, line10

jal duyet

la \$k0, line11

jal duyet

la \$k0, line12

jal duyet

la \$k0, line13

jal duyet

la \$k0, line14

jal duyet

la \$k0, line15

jal duyet

la \$k0, line16

jal duyet

add \$ra,\$0,\$k1

jr \$ra

tiếp tục xét màu của kí tự tiếp theo

duyet: add \$t1,\$k0,\$t0

address i # duyet tung kí tự trong khoảng \$a2 - \$a3

lb \$t2,0(\$t1)

value i

beq \$t0,\$a3, end_duyet

đến \$a3 dừng lại

beq \$t2,\$t7,change

thay đổi màu

jump: addi \$t0,\$t0,1

j duyet

change: sb \$v1, 0(\$t1)

j jump

tiếp tục duyet kí tự sau

end_duyet: add \$t0,\$a2,\$0

jr \$ra

printf: # thực hiện in từng dòng

li \$v0,4

li \$t0,0

la \$a0,line1

syscall

la \$a0,line2

syscall

la \$a0,line3

syscall

la \$a0,line4

syscall

la \$a0,line5

syscall

la \$a0,line6

syscall

la \$a0,line7

syscall

la \$a0,line8

syscall

la \$a0,line9

syscall

la \$a0,line10

syscall

la \$a0,line11

syscall

la \$a0,line12

syscall

la \$a0,line13

syscall

la \$a0,line14

syscall

la \$a0,line15

syscall

la \$a0,line16

syscall

li \$t0,0

li \$t1,0

j loop

DAO: li \$t0,0

khoi tao i = 0

```

loop2:   add $t1,$a0,$t0           # duyệt tung phần tử

        #định tiên các kí tự lên 43 o nhỏ và cả kí tự từ 43 trở lên đi chuyển xuống 43 o nhỏ

        lb $t2,0($t1)

        lb $t3,43($t1)            # đổi vị trí 2 kí tự cách nhau 43 o nhỏ

        sb $t3,0($t1)

        sb $t2,43($t1)

        li $t1,21                 # thực hiện với 21 kí tự (do kí tự lớn nhất cần đảo
la 21)                                là 21)

        beq $t0,$t1,end_loop2

        addi $t0,$t0,1             # i = i + 1

        j loop2

end_loop2: jr $ra                  # quay lại duyệt dòng tiếp theo

END_DAO:

XU_LY_VIEN:      add $t1,$a1,$t0    # thực hiện duyệt tung kí tự - $a1 là biến lưu trữ
line i

        lb $a0,0($t1)

        li $t1,10                  # t1 = '\n'

        beq $a0,$t1,end_xu_ly_vien  # if($t1 == "\n") end_xy_ly_vien dòng i

        li $t1,48

        slt $t2,$a0,$t1             # $t2 = $a0 < 48 ? 1:0

        li $t1,57

        slt $t3,$t1,$a0             # $t3 = 57 < $a0 ? 1:0

        or $t1,$t2,$t3              # if(48<= $a0 <= 57) print " "

        beq $t1,$0,print_32

        li $v0,11                   # if($a0<48 || $a0 >57) print $a0

        syscall

        j jump_xu_ly_vien           # quay lại vòng lặp

print_32:       li $v0,11            # print " "

                li $a0,32

```

```

        syscall

jump_xu_ly_vien:addi $t0,$t0,1                                # quay lai vong lap

        j XU_LY_VIEN

end_xu_ly_vien: li $v0,11                                     # print "\n"

        syscall

        li $t0,0                                             # khoi tao lai i = 0

        jr $ra                                               # quay lai duyet dong tiep theo

khong_co_chuc_nang:  li $v0,4                                # thong bao loi

                    la $a0,thong_bao

                    syscall

j loop

end:

```