BÁO CÁO CUỐI KÌ MÔN THỰC HÀNH KIẾN TRÚC MÁY TÍNH

Mã Lớp: 130999

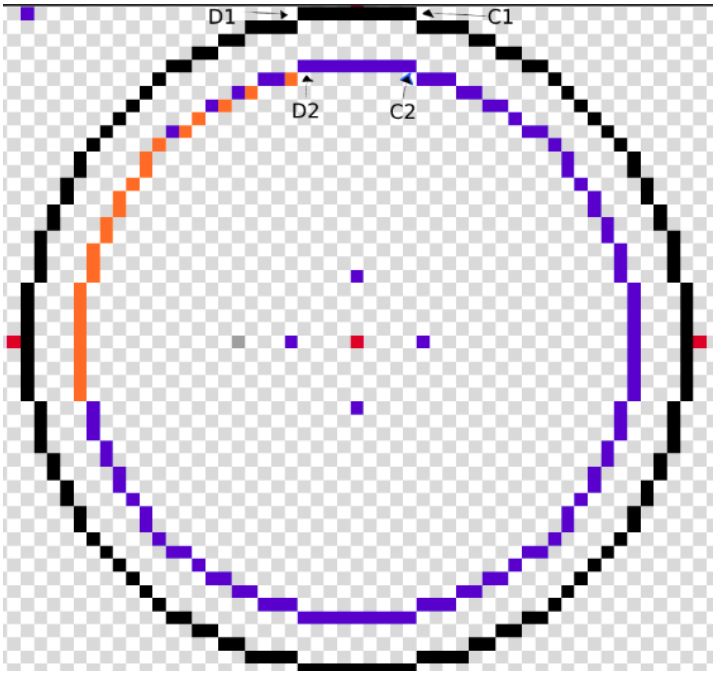
BTCK 2:

Mô tả bài toán:

Vẽ một quả bóng trên màn hình mô phỏng Bitmap. Quả bóng di chuyển phụ thuộc vào phím người dùng bấm vào bộ giả lập MMIO. Nhấn w, s, a, d lần lượt di chuyển lên, xuống, trái, phải. Nếu chạm vào biên thì quả bóng bật lại. Tốc độ không đổi.

Phương pháp giải bài toán:

- Đầu tiên phải vẽ quả bóng, bằng cách tô màu đường tròn bằng màu vàng. Tức là ta tô màu một loạt các pixel trên màn hình mô phỏng Bitmap và tạo ra được một đường tròn màu vàng.
    + Cách tạo đường tròn:



    + Ta cần tô màu 2 viền tròn bên trên và các điểm bên trong bằng màu vàng, ta sẽ thu được đường tròn vàng
        - Vòng ngoài: từ dòng 1 đến 51
        - Vòng trong: từ dòng 5 đến 46
    + Trong hình ta thấy:
        - D1 là điểm đầu tiên bên trái của đường tròn bên ngoài
        - D2 là điểm cuối cùng bên phải của đường tròn bên ngoài
        - C1 là điểm đầu tiên bên trái của đường tròn bên trong
        - C2 là điểm cuối cùng bên phải của đường trong bên trong
    + Cách tô: Tô từ trên xuống dưới, tô từng dòng, tô từng pixel từ trái sang phải
        - Từ dòng 1 -> 5 : Tô màu từ D1 đến C1
        - Từ dòng 6 -> 46 : Tô màu từ D1 đến D2 và từ C2 đến C1
        - Từ 47 -> 51: Tô màu từ D1 đến C1

- Tiếp theo là khi người dùng nhập ký tự (w, a, s, d) vào trong MMIO thì quả bóng sẽ di chuyển.
- Để di chuyển quả bóng thì ta sẽ vẽ quả bóng ở một vị trí mới và xóa quả bóng ở vị trí cũ (chính là tô màu quả bóng cũ giống màu nền). Vị trí mới của quả bóng mới tùy thuộc vào người dùng nhập ký tự nào.
    + Vị trí mới sẽ được xác định như sau:
        - Nếu là w: Vị trí mới = Vị trí cũ - 512. Tức là dịch lên một dòng, vì một dòng có 512 pixel. Khi trừ đi 512 thì sẽ dịch lên một dòng.
        - Nếu là s: Vị trí mới = Vị trí cũ + 512. Tức là dịch xuống một dòng.
        - Nếu là a: Vị trí mới = Vị trí cũ - 1. Dịch trái một cột
        - Nếu là d: Vị trí mới = Vị trí cũ + 1. Dich phải một cột
- Khi chạm vào viền thì đập ngược lại.
    + Chẳng hạn nếu đang di chuyển lên mà chạm viền thì chuyển sang chế độ di chuyển xuống và ngược lại
    + Còn nếu di chuyển sang trái mà chạm viền thì chuyển sang phải và ngược lại

Code MIPS assembly và giải thích code:

```
1  .eqv MONITOR_SCREEN 0x10010000 #Dia chi bat dau cua bo nho man hinh
2
3  .eqv TopHead_1 118012   # D1 ban dau
4  .eqv TopTeal_1 118020   # C1 ban dau
5
6  .eqv TopHead_2 120060   # D2 ban dau
7  .eqv TopTeal_2 120068   # C2 ban dau
8
9
10 .eqv KEY_CODE 0xFFFF0004        # ASCII code from keyboard, 1 byte
11 .eqv KEY_READY 0xFFFF0000       # =1 if has a new keycode ?
12  # Auto clear after lw
13
14 .eqv DISPLAY_CODE 0xFFFF000C    # ASCII code to show, 1 byte
15 .eqv DISPLAY_READY 0xFFFF0008   # =1 if the display has already to do
16  # Auto clear after sw
```

Giải thích code bên trên:

- Khai báo các hằng và gán giá trị
- Dòng 1, 10, 11, 14, 15 : là các hằng mang giá trị địa chỉ đầu vào và ra của các công cụ bitmap, bộ giả lập MMIO
- Dòng 3, 4, 6, 7: là vị trí bắt đầu vẽ quả bóng

```
19 .text
20
21 # ---------------------------------------------------------------------------------------
22 # TAC DUNG CUA CAC THANH GHI TRONG CODE TAO HINH TRON
23 #----------------------------------------------------------------------------------------
24 # s0 : dem so dong
25 # s1, s2 : 2 bien trong ham tinh cal
26 # s3, s4 : 2 bien trong ham tinh calcu
27 # s5, s6 : 2 bien trong ham to mau
28 # t8: dia chi diem dau stack
29 # t9: dia chi diem cuoi stack
30 # ---------------------------------------------------------------------------------------
```

Giải thích code bên trên:

- Em thường comment lại những thanh ghi được sử dụng 1 chức năng xuyên suốt một đoạn code lớn lên trên đầu để dễ đối chiếu và xem xét.

```
33 # ------------------------------------------------------------------------------------
34 #       CODE TAO HINH TRON
35 # ------------------------------------------------------------------------------------
36         li $k0, MONITOR_SCREEN  # Nap dia chi bat dau cua man hinh
37         li $s0, 1               # So dong
38         add $t8, $sp, $zero     # Luu dia chi diem dau stack
```

Giải thích code bên trên:

- Gán k0 = địa chỉ bắt đầu của màn hình bitmap
- Cho biến số dòng s0 = 1
- Lưu địa chỉ điểm đầu stack vào t8

```
41 # Ham xet tung dong
42 main:
43
44 main_root:
45         # cac bien dau tien cho ham cal*
46         li $s1, TopHead_1
47         li $s2, TopTeal_1
48
49         # cac bien dau tien cua ham calcu*
50         li $s3, TopHead_2
51         li $s4, TopTeal_2
52
53         # To mau dong dau tien
54         li $s5, TopHead_1
55         li $s6, TopTeal_1
56         jal color_main
57
58         addi $s0, $s0, 1        # tang so dong
```

Giải thích code bên trên:

- Gán D1 vào s1, D2 vào s2, C1 vào s3, C2 vào s4
- Gán D1 vào s5, D2 vào S6 để tô màu từ trái (D1) sang phải (D2)
- Sau mỗi lần tô màu xong 1 dòng thì tăng giá trị biến s0 (biến đếm số dòng)

```
60  main_ele_circle_1:
61          # dong 2
62          beq $s0, 2, main_ele_1
63          # dong 3 -> 5
64          slti $t1, $s0, 6
65          li $t3, 2
66          slt $t2, $t3, $s0
67          add $t4, $t1, $t2
68          beq $t4, 2, main_ele_3
69          # dong 6 -> 12
70          slti $t1, $s0, 13
71          li $t3, 5
72          slt $t2, $t3, $s0
73          add $t4, $t1, $t2
74          beq $t4, 2, main_ele_4
75          # dong 13
76          beq $s0, 13, main_ele_5
77          # dong 14
78          beq $s0, 14, main_ele_4
79          # dong 15
80          beq $s0, 15, main_ele_5
81          # dong 16
82          beq $s0, 16, main_ele_4
83          # dong 17
84          beq $s0, 17, main_ele_5
85          # dong 18
86          beq $s0, 18, main_ele_4
87          # dong 19 -> 21
88          slti $t1, $s0, 22
89          li $t3, 18
90          slt $t2, $t3, $s0
91          add $t4, $t1, $t2
92          beq $t4, 2, main_ele_5
93          # dong 22
94          beq $s0, 22, main_ele_4
95          # dong 23 -> 30
96          slti $t1, $s0, 31
97          li $t3, 22
98          slt $t2, $t3, $s0
99          add $t4, $t1, $t2
100         beq $t4, 2, main_ele_5
```

```
102
103         # dong 31
104         beq $s0, 31, convert_1
105 main_ele_circle_1_back:
106         beq $s0, 31, main_ele_4
107         # dong 32 -> 34
108         slti $t1, $s0, 35
109         li $t3, 31
110         slt $t2, $t3, $s0
111         add $t4, $t1, $t2
112         beq $t4, 2, main_ele_5
113         # dong 35
114         beq $s0, 35, main_ele_4
115         # dong 36
116         beq $s0, 36, main_ele_5
117         # dong 37
118         beq $s0, 37, main_ele_4
119         # dong 38
120         beq $s0, 38, main_ele_5
121         # dong 39
122         beq $s0, 39, main_ele_4
123         # dong 40
124         beq $s0, 40, main_ele_5
125         # dong 41 -> 47
126         slti $t1, $s0, 48
127         li $t3, 40
128         slt $t2, $t3, $s0
129         add $t4, $t1, $t2
130         beq $t4, 2, main_ele_4
131         # dong 48 -> 50
132         slti $t1, $s0, 51
133         li $t3, 47
134         slt $t2, $t3, $s0
135         add $t4, $t1, $t2
136         beq $t4, 2, main_ele_3
137         # dong 51
138         beq $s0, 51, main_ele_1
```

Giải thích code bên trên:

- **main_ele_circle_1**: Ở đây ta dùng biến đếm dòng s0 để xác định dòng cần tô màu và chuyển đến hàm xác định điểm D1, C1 của dòng cần tô màu sau đó chuyển đến hàm **main_ele_circle_2.**

```
140 main_ele_circle_2:
141         # dong 6
142         beq $s0, 6, main_element_1
143         # dong 7
144         beq $s0, 7, main_element_3
145         # dong 8, 9
146         beq $s0, 8, main_element_2
147         beq $s0, 9, main_element_2
148         # dong 10
149         beq $s0, 10, main_element_1
150         # dong 11
151         beq $s0, 11, main_element_2
152         # dong 12
153         beq $s0, 12, main_element_1
154         # dong 13
155         beq $s0, 13, main_element_0
156         # dong 14
157         beq $s0, 14, main_element_1
```

```
158         # dong 15
159         beq $s0, 15, main_element_1
160         # dong 16
161         beq $s0, 16, main_element_0
162         # dong 17
163         beq $s0, 17, main_element_1
164         # dong 18
165         beq $s0, 18, main_element_0
166         # dong 19
167         beq $s0, 19, main_element_1
168         # dong 20, 21
169         beq $s0, 20, main_element_0
170         beq $s0, 21, main_element_0
171         # dong 22
172         beq $s0, 22, main_element_1
173         # dong 23 -> 30
174         slti $t1, $s0, 31
175         li $t3, 22
176         slt $t2, $t3, $s0
177         add $t4, $t1, $t2
178         beq $t4, 2, main_element_0
```

```
180         # dong 31
181         beq $s0, 31, convert_2
182 main_ele_circle_2_back:
183         beq $s0, 31, main_element_1
184         # dong 32, 33
185         beq $s0, 32, main_element_0
186         beq $s0, 33, main_element_0
187         # dong 34
188         beq $s0, 34, main_element_1
189         # dong 35
190         beq $s0, 35, main_element_0
191         # dong 36
192         beq $s0, 36, main_element_1
193         # dong 37
194         beq $s0, 37, main_element_0
195         # dong 38, 39
196         beq $s0, 38, main_element_1
197         beq $s0, 39, main_element_1
198         # dong 40
199         beq $s0, 40, main_element_0
```

```
200         # dong 41, 42
201         beq $s0, 41, main_element_1
202         beq $s0, 42, main_element_1
203         # dong 43
204         beq $s0, 43, main_element_2
205         # dong 44
206         beq $s0, 44, main_element_1
207         # dong 45, 46
208         beq $s0, 45, main_element_2
209         beq $s0, 46, main_element_2
```

Giải thích code bên trên:

- **main_ele_circle_2**: Ở đây ta dùng biến đếm dòng s0 để xác định dòng cần tô màu và chuyển đến hàm xác định điểm D2, C2 của dòng cần tô màu. Đến đây ta đã xác định được D1, C1, D2, C2 => chuyển đến hàm tô màu.

```
211 main_color:
212         # dong 1 -> 5
213         slti $t1, $s0, 6
214         li $t3, 0
215         slt $t2, $t3, $s0
216         add $t4, $t1, $t2
217         beq $t4, 2, main_color_1        # neu thuoc dong 1 -> 5 thi nhay den nhan
218         # dong 6 -> 46
219         slti $t1, $s0, 47
220         li $t3, 5
221         slt $t2, $t3, $s0
222         add $t4, $t1, $t2
223         beq $t4, 2, main_color_2        # neu thuoc dong 6 -> 46 thi nhay den nhan
224         # dong 47 -> 51
225         slti $t1, $s0, 52
226         li $t3, 46
227         slt $t2, $t3, $s0
228         add $t4, $t1, $t2
229         beq $t4, 2, main_color_1        # neu thuoc dong 47 -> 51 thi nhay den nhan
```

Giải thích code bên trên:

- **main_color:** dùng s0 để xác định dòng hiện tại cần tô màu và chuyển đến hàm tô màu thích hợp
  - + Từ 1-> 5 và 47 -> 51: Chỉ cần tô màu từ D1 đến C1
  - + Từ 6 -> 46: Cần tô từ D1 đến D2 và C2 đến C1. Vì khoảng trống ở bên trong đường tròn không cần tô, tức là không cần tô D2 đến C2

```
231 main_color_1:
232         # To mau tu D1 -> C1
233         add $s5, $s1, $zero     # s5 = s1
234         add $s6, $s2, $zero     # s6 = s2
235         jal color_main
236         # Sau khi to mau xong thi tang so dong len
237         j main_raise
```

Giải thích code bên trên:

- **main_color-1:** Tô màu từ D1 đến C1
  => Gán D1, C1 vào 2 biến s5, s6 (2 tham số của hàm tô màu) sau đó chuyển đến hàm tô màu

```
239 main_color_2:
240         # To mau D1 -> D2
241         add $s5, $s1, $zero     # s5 = s1
242         add $s6, $s3, $zero     # s6 = s3
243         jal color_main
244         # To mau C2 -> C1
245         add $s5, $s4, $zero     # s5 = s4
246         add $s6, $s2, $zero     # s6 = s2
247         jal color_main
248         # Sau khi to mau xong thi tang so dong len
249         j main_raise
```

Giải thích code bên trên:

- **main_color-2:** Tô màu từ D1 đến D2 và từ C2 đến C1
  => Gán D1, D2 vào 2 biến s5, s6 (2 tham số của hàm tô màu) sau đó chuyển đến hàm tô màu

=> Sau đó gán C2, C1 vào 2 biến s5, s6 (2 tham số của hàm tô màu) sau đó chuyển đến hàm tô màu

```
251
252 # Ham tang so dong
253 main_raise:
254         beq $s0, 51, main_out   # neu den dong 51 thi thoat main
255         addi $s0, $s0, 1        # tang so dong
256         j main_ele_circle_1
```

Giải thích code bên trên:

- **main_raise:** Kiểm tra nếu dòng vừa tô là dòng 51 thì thoát ra khỏi hàm main nếu không thì tăng số dòng và tiếp tục tô màu.

```
258 # Ham mo rong nhay den Cal* va quay lai main_ele_circle_2
259 main_ele_1:
260         jal Cal_4
261         j main_ele_circle_2
262 main_ele_3:
263         jal Cal_2
264         j main_ele_circle_2
265 main_ele_4:
266         jal Cal_1
267         j main_ele_circle_2
268 main_ele_5:
269         jal Cal_0
270         j main_ele_circle_2
```

Giải thích code bên trên:

- **main_ele_1, main_ele_3, main_ele_4, main_ele_5:** Chuyển đến các cách tính D1, C1 tương ứng, sau đó nhảy đến **main_ele_circle_2**

```
271
272 # Ham mo rong nhay den Calcu* va quay lai main_raise de tang so dong
273 main_element_0:
274         jal Calcu_0
275         j main_color
276 main_element_1:
277         jal Calcu_1
278         j main_color
279 main_element_2:
280         jal Calcu_2
281         j main_color
282 main_element_3:
283         jal Calcu_3
284         j main_color
285
286
```

Giải thích code bên trên:

- **main_element_0, main_element_1, main_element_2, main_element_3:** Chuyển đến các cách tính D2, C2 tương ứng, sau đó nhảy đến **main_color**

```
288 # Ham tinh D = s1, C = s2
289 Cal_0:
290         addi $s1, $s1, 512
291         addi $s2, $s2, 512
292         jr $ra
293
294 # Ham tinh D = s1, C = s2
295 Cal_1:
296         addi $s1, $s1, 511
297         addi $s2, $s2, 513
298         jr $ra
299
300 # Ham tinh D = s1, C = s2
301 Cal_2:
302         addi $s1, $s1, 510
303         addi $s2, $s2, 514
304         jr $ra
305
306 # Ham tinh D = s1, C = s2
307 Cal_4:
308         addi $s1, $s1, 508
309         addi $s2, $s2, 516
310         jr $ra
```

Giải thích code bên trên:

- **Cal_0, Cal_1, Cal_2, Cal_4:** Tính D1, C1 của dòng cần tô màu tiếp theo

```
312 # Ham tinh D = s3, C = s4
313 Calcu_0:
314         addi $s3, $s3, 512
315         addi $s4, $s4, 512
316         jr $ra
317
318 # Ham tinh D = s3, C = s4
319 Calcu_1:
320         addi $s3, $s3, 511
321         addi $s4, $s4, 513
322         jr $ra
323
324 # Ham tinh D = s3, C = s4
325 Calcu_2:
326         addi $s3, $s3, 510
327         addi $s4, $s4, 514
328         jr $ra
329
330 # Ham tinh D = s3, C = s4
331 Calcu_3:
332         addi $s3, $s3, 509
333         addi $s4, $s4, 515
334         jr $ra
```

Giải thích code bên trên:

- **Calcu_0, Calcu_1, Calcu_2, Calcu_3:** Tính D2, C2 của dòng cần tô màu tiếp theo

```
# Ham to mau cac pixel tu s5 -> s6
color_main:
        slt $t5, $s5, $s6       # neu s5 > s6 thi can convert s5 va s6
        beqz $t5, convert_3
color_main_back:
        add $t1, $s5, $zero     # t1 = s5
color_ele:
        add $t6, $t1, $zero
        sw $t6, 0($sp)          # Loop gan toan bo cac diem pixel vao stack
        addi $sp, $sp, -4       # Den ngan tiep theo

        mul $t3, $t1, 4         # t3 = t1 * 4
        add $t4, $k0, $t3       # t4 = k0 + t1 * 4
        li $t2, 0x00FFFF00      # t2 = YELLOW
        sw $t2, 0($t4)          # k0 = YELLOW
        beq $t1, $s6, color_out # neu t1 = s6 thi thoat khoi ham to mau
        addi $t1, $t1, 1        # t1 = t1 + 1
        j color_ele
color_out:
        jr $ra
```

Giải thích code bên trên:

- **color_main:** bắt đầu tô màu. Tô từng điểm từ s5 đến s6
    + Kiểm tra nếu s5 > s6 thì hoán đổi s5 và s6 cho nhau
- **color_main_back:** điểm quay lại sau khi hoán đổi
- **color_ele:** Bắt đầu tô màu từng pixel và lưu các pixel đó vào trong ngăn xếp.
- **color_out:** Thoát khỏi hàm tô màu

```
# Doi vi tri tu s1 thanh s2 va nguoc lai
convert_1:
        add $t1, $s1, $zero     # t1 = s1
        add $s1, $s2, $zero     # s1 = s2
        add $s2, $t1, $zero     # s2 = t1
        j main_ele_circle_1_back


# Doi vi tri tu s3 thanh s4 va nguoc lai
convert_2:
        add $t1, $s3, $zero     # t1 = s3
        add $s3, $s4, $zero     # s3 = s4
        add $s4, $t1, $zero     # s4 = t1
        j main_ele_circle_2_back

# Doi vi tri tu s5 thanh s6 va nguoc lai
convert_3:
        add $t1, $s5, $zero     # t1 = s5
        add $s5, $s6, $zero     # s5 = s6
        add $s6, $t1, $zero     # s6 = t1
        j color_main_back
```

Giải thích code bên trên:

- **convert_1:** Hàm hoán đổi s1 và s2
- **convert_2:** Hàm hoán đổi s3 và s4
- **convert_3:** Hàm hoán đổi s5 và s6

```
379 # Ham thoat khoi chuong trinh main
380 main_out:
381         add $t9, $sp, $zero     # Luu dia chi diem cuoi stack
382 # ------------------------------------------------------------------------------------
```

Giải thích code bên trên:

- **main_out:** Sau khi vẽ xong dòng 51 thì lưu điểm cuối của stack và đến phần di chuyển quả bóng

```
385 #---------------
386 #GIAI PHONG BO NHO
387 li $s1, 0
388 li $s2, 0
389 li $s3, 0
390 li $s4, 0
391 li $s5, 0
392 li $s6, 0
393 li $t1, 0
394 li $t2, 0
395 li $t3, 0
396 li $t4, 0
397 li $t6, 0
398 #---------------
```

Giải thích code bên trên:

- Giải phóng bộ nhớ với mong muốn code sẽ chạy nhanh hơn.

```
403 # ------------------------------------------------------------------------------------
404 # TAC DUNG CUA CAC THANH GHI TRONG CODE TAO CO CHE DI CHUYEN CUA HINH TRON
405 #-------------------------------------------------------------------------------------
406 # s0: gia tri diem pixel lay ra tu stack
407 # s1: gia tri tuong ung voi che do di chuyen (W, D, S, A)
408 #
409 #
410 # t8: dia chi diem dau stack
411 # t9: dia chi diem cuoi stack
412 # ------------------------------------------------------------------------------------
413
414
415 # ------------------------------------------------------------------------------------
416 #       CODE TAO CO CHE DI CHUYEN CUA HINH TRON
417 # ------------------------------------------------------------------------------------
418
419
420         li $a3, KEY_CODE        # ASCII code from keyboard, 1 byte
421         li $k1, KEY_READY       # =1 if has a new keycode ?
422                                 # Auto clear after lw
423
424         li $a0, DISPLAY_CODE    # ASCII code to show, 1 byte
425         li $a1, DISPLAY_READY   # =1 if the display has already to do
426                                 # Auto clear after sw
```

Giải thích code bên trên:

- Nạp địa chỉ nhận ký tự từ bàn phím (KEY_CODE) vào a3
- Nạp địa chỉ kiểm tra có ký tự mới được nhập không (KEY_READY) vào k1
- Nạp địa chỉ hiển thị ký tự vào a0
- Nạp địa chỉ sẵn sàng hiển thị vào a1

```
loop:
        nop
#-----------------------------------------------------
WaitForKey:
        lw $t1, 0($k1) # $t1 = [$k1] = KEY_READY
        nop
        beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling
        nop
#-----------------------------------------------------
ReadKey:
        lw $t0, 0($a3) # $t0 = [$k0] = KEY_CODE
        nop
#-----------------------------------------------------
WaitForDis:
        lw $t2, 0($a1) # $t2 = [$s1] = DISPLAY_READY
        nop
        beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling
        nop
#-----------------------------------------------------
ShowKey:
        sw $t0, 0($a0) # show key
        nop
```

Giải thích code bên trên:

- Vòng lặp để kiểm tra người dùng có nhập ký tự vào không

```
CheckKey:
        beq $t0, 'w', key_W
        beq $t0, 'd', key_D
        beq $t0, 's', key_S
        beq $t0, 'a', key_A
        bne $s1, 0, check_border
        j WaitForKey
```

Giải thích code bên trên:

- **CheckKey:** Kiểm tra xem ký tự vừa nhập là ký tự nào. Nếu là w, s, d, a thì tiến hành di chuyển đường tròn theo quy ước và sau đó kiểm tra điều kiện viền. Nếu là ký tự khác thì quay lại chờ ký tự mới.

```

# Xac dinh huong di chuyen
# len
key_W:
        li $s1, -512
        j convert_color
# trai
key_A:
        li $s1, -1
        j convert_color
# phai
key_D:
        li $s1, 1
        j convert_color_2
# xuong
key_S:
        li $s1, 512
        j convert_color_2
```

Giải thích code bên trên:

- **Key_W, Key_A, Key_D, Key_S:** Gán giá trị thích hợp cho s1, để khi cộng s1 vào vị trí cũ sẽ ra vị trí mới

```
# Dao chieu di chuyen
convert_color_back:
        mul $s1, $s1, -1        # dao chieu khi va phai bien
        beq $s1, -512, convert_color
        beq $s1, -1, convert_color
        beq $s1, 512, convert_color_2
        beq $s1, 1, convert_color_2
```

Giải thích code bên trên:

- **Convert_color_back:** Đảo chiều di chuyển khi gặp viền, chỉ cần nhân s1 với -1 ta có thể đảo ngược chiều di chuyển.

```
# Ham to mau cac pixel tu dau stack -> cuoi stack
convert_color:
        add $sp, $t8, $zero     # lay diem dau stack

convert_color_ele:
        lw $s0, 0($sp)          # lay vi tri pixel tu ngan xep
        # Convert pixel vang -> den
        mul $t3, $s0, 4         # t3 = s0 * 4
        add $t4, $k0, $t3       # t4 = k0 + s0 * 4
        li $t2, 0x0             # t2 = DARK
        sw $t2, 0($t4)          # k0 = DARK

        # Luu vao stack
        add $s0, $s0, $s1       # Xac dinh pixel can to mau
        add $s6, $s0, $zero
        sw $s6, 0($sp)          # gan lai vao ngan stack vua lay

        # convert pixel den -> vang
        mul $t3, $s0, 4         # t3 = s0 * 4
        add $t4, $k0, $t3       # t4 = k0 + s0 * 4
        li $t2, 0x00FFFF00      # t2 = YELLOW
        sw $t2, 0($t4)          # k0 = YELLOW

        addi $sp, $sp, -4       # den ngan nho tiep theo
        beq $sp, $t9, check_new_key     # neu het stack thi khong to mau nua
        j convert_color_ele     # lap de to mau pixel tiep theo
```

Giải thích code bên trên:

- **convert_color:** Bắt đầu di chuyển quả bóng bằng cách đổi màu vị trí cũ về màu nền và màu vị trí mới bằng màu vàng. Đổi màu từ đầu ngăn xếp đến cuối ngăn xếp (từ trên xuống dưới của đường tròn)
- **convert_color_ele:** Lần lượt lấy các vị trí từ ngăn xếp ra, đổi màu về màu nền, cộng thêm s1 để tạo thành vị trí mới, lưu vào ngăn xếp và tô màu vàng cho vị trí mới. Tiếp tục cho đến khi hết ngăn xếp

```
512 # Ham to mau cac pixel tu cuoi stack -> dau stack
513 convert_color_2:
514         add $sp, $t9, $zero
515         addi $sp, $sp, 4
516
517 convert_color_ele_2:
518         lw $s0, 0($sp)          # lay vi tri pixel tu ngan xep
519         # Convert pixel vang -> den
520         mul $t3, $s0, 4         # t3 = s0 * 4
521         add $t4, $k0, $t3       # t4 = k0 + s0 * 4
522         li $t2, 0x0             # t2 = DARK
523         sw $t2, 0($t4)          # k0 = DARK
524
525         # Luu vao stack
526         add $s0, $s0, $s1       # Xac dinh pixel can to mau
527         add $s6, $s0, $zero
528         sw $s6, 0($sp)          # gan lai vao ngan stack vua lay
529
530         # convert pixel den -> vang
531         mul $t3, $s0, 4         # t3 = s0 * 4
532         add $t4, $k0, $t3       # t4 = k0 + s0 * 4
533         li $t2, 0x00FFFF00      # t2 = YELLOW
534         sw $t2, 0($t4)          # k0 = YELLOW
```

```
536         beq $sp, $t8, check_new_key     # neu het stack thi khong to mau nua
537         addi $sp, $sp, 4        # den ngan nho tiep theo
538         j convert_color_ele_2   # lap de to mau pixel tiep theo
```

Giải thích code bên trên:

- **convert_color_2:** Giống **convert_color** nhưng là tô từ cuối ngăn xếp đến đầu ngăn xếp (Tô từ dưới lên trên của đường tròn)
- **convert_color_ele_2:** Giống như **convert_color_ele.** Lần lượt lấy các vị trí từ ngăn xếp ra, đổi màu về màu nền, cộng thêm s1 để tạo thành vị trí mới, lưu vào ngăn xếp và tô màu vàng cho vị trí mới. Tiếp tục cho đến khi hết ngăn xếp

```
540 #check new key tu keyboard
541 check_new_key:
542         lw $t1, 0($k1) # $t1 = [$k1] = KEY_READY
543         nop
544         bne $t1, $zero, ReadKey # if $t1 == 0 then Polling
545         nop
```

Giải thích code bên trên:

- **check_new_key:** Kiểm tra xem có ký tự mới được nhập hay không

```
547 # bat dau check bien
548 check_border:
549         add $sp, $t8, $zero     # con tro sp tro vao dau stack
550         lw $s0, 0($sp)          # lay vi tri pixel tu ngan xep dau tien
551
552         beq $s1, -512, check_row_top
553         beq $s1, 512, check_row_bottom
554         beq $s1, -1, check_col_left
555         beq $s1, 1, check_col_right
556         beq $s1, 0, WaitForKey
```

Giải thích code bên trên:

- **check_border:** Lấy vị trí đầu tiên trên cùng bên trái của hình tròn để kiểm tra điều kiện viền (gọi là vị trí giới hạn biên s0)
- Sau đó tùy vào giá trị s1 mà xác định viền nào cần kiểm tra (viền trên, dưới, trái hay phải)

```
# check s0 co thuoc 22 -> 483
check_row_top:
        slti $t1, $s0, 484
        li $t3, 21
        slt $t2, $t3, $s0
        add $t4, $t1, $t2
        beq $t4, 2, convert_color_back  # neu thuoc khoang kiem tra thi dao chieu di chuyen
        j convert_color                 # neu khong thi tiep tuc to mau

# check s0 co thuoc 236054 -> 236515
check_row_bottom:
        add $t5, $s0, $zero
        li $t6, -236000
        add $t5, $t5, $t6

        slti $t1, $t5, 516
        li $t3, 53
        slt $t2, $t3, $t5
        add $t4, $t1, $t2
        beq $t4, 2, convert_color_back  # neu thuoc khoang kiem tra thi dao chieu di chuyen
        j convert_color_2                       # neu khong thi tiep tuc to mau
```

```
# check bien ben phai
check_col_right:
        add $t5, $s0, $zero
        addi $t5, $t5, 29
        div $t5, $t5, 512
        mfhi $t6
        bne $t6, 0, convert_color_2     # neu so du khac 0 thi chua den bien

        slti $t1, $t5, 463
        li $t3, 0
        slt $t2, $t3, $t5
        add $t4, $t1, $t2
        beq $t4, 2, convert_color_back  # neu thuoc khoang kiem tra thi dao chieu di chuyen
        j convert_color_2               # neu khong thi tiep tuc to mau

# check bien ben trai
check_col_left:
        add $t5, $s0, $zero
        addi $t5, $t5, 490
        div $t5, $t5, 512
        mfhi $t6
        bne $t6, 0, convert_color       # neu du khac 0 thi chua den bien

        slti $t1, $t5, 463
        li $t3, 0
        slt $t2, $t3, $t5
        add $t4, $t1, $t2
        beq $t4, 2, convert_color_back  # neu thuoc khoang kiem tra thi dao chieu di chuyen
        j convert_color         # neu khong thi tiep tuc to mau
#-----------------------------------------------------
```

Giải thích code:

- Với những vị trí giới hạn biên của s0 được tính sẵn trong 4 trường hợp:
  - + Trên: 22 -> 483
  - + Dưới: 236054 -> 236515
  - + Trái: Khi cộng thêm 29 thì sẽ chia hết cho 512
  - + Phải: Khi cộng thêm 490 thì sẽ chia hết cho 512

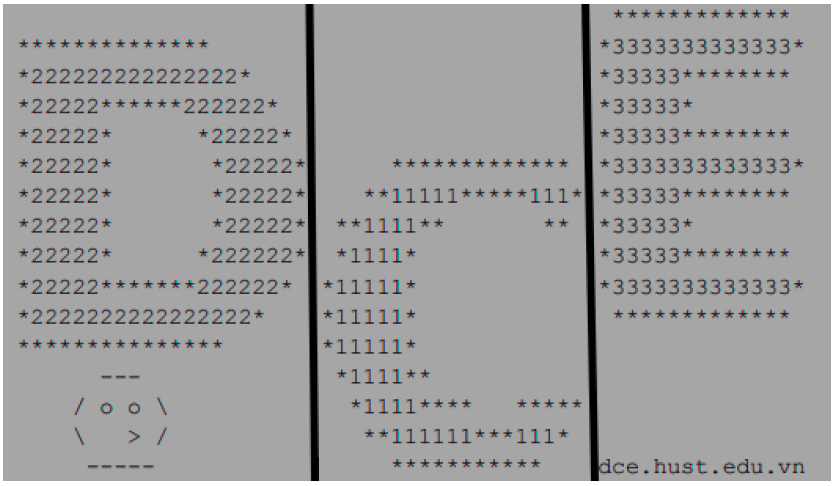
BTCK 9:

Mô tả bài toán:

- Cho hình chữ DCE với viền là các ký tự * và màu là các con số
- Hãy hoàn thành 4 yêu cầu sau:
  + Hiển thị hình ảnh ra giao diện console
  + Sửa ảnh các chữ cái chỉ còn lại viền, không có màu ở giữa và hiển thị
  + Hoán đổi các chữ thành ECD và hiển thị
  + Nhập các màu từ bàn phím lần lượt cho D, C, E rồi hiển thị

Phương pháp giải bài toán:

- Chia ảnh ra làm 3 phần lần lượt là phần của D, C và E



- Tiến hành lưu từng phần vào trong các biến: phần D lưu vào trong biến image_D, phần C lưu vào trong image_C, phần E lưu vào trong image_E.
  + Cách lưu:
    - Lưu các dòng vào trong biến, các dòng ngăn cách bởi ký tự '\n', dùng ký tự '?' để biểu thị việc kết thúc một phần (tức là hết ký tự)
    - Các chỗ trống trong ảnh cần sử dụng chuỗi ký tự space (' ') để thể hiện
- Yêu cầu thứ nhất:
  + Để in ra hình ảnh ra giao diện console, ta dựa vào cách lưu vào trong biến image_D, image_C, image_E.
  + Ý tưởng:
    - In lần lượt từng dòng trong image_D, image_C, image_E cho đến khi in hết toàn bộ ký tự ở trong 3 biến
    - Nói cách khác chính là in dòng thứ nhất của 3 biến image_D, image_C, image_E rồi xuống dòng, sau đó in dòng thứ hai của 3 biến image_D, image_C, image_E rồi xuống dòng, sau đó … Đến khi hết gặp ký tự '?' thể hiện việc hết ký tự
- Yêu cầu thứ hai:
  + Để sửa ảnh các chữ cái chỉ còn lại viền, không có màu ở giữa ta chỉ việc thay thế ký tự màu bằng ký tự space (' ')
  + Vẫn dùng cách duyệt như trong yêu cầu thứ nhất

    + Thêm một bước kiểm tra ký tự màu và thay thế
- Yêu cầu thứ ba:
    + Giống như yêu cầu thứ nhất ta chỉ thay thứ tự in biến thành image_E, image_C, image_D.
- Yêu cầu thứ tư:
    + Yêu cầu người dùng nhập màu cho chữ
    + Sau đó kiểm tra tính hợp lệ của màu vừa nhập
    + Giống như yêu cầu thứ hai, kiểm tra ký tự màu và thay thế bằng ký tự thích hợp người dùng nhập đã nhập

Code MIPS assembly và giải thích code:

```
1  .data
2          image_D: .asciiz "                      \n **************       \n *222222222222222*
3          image_C: .asciiz "                    \n                    \n                      \n
4          image_E: .asciiz "  ************* \n *3333333333333*\n *33333******** \n *33333*
5
6          color_D: .asciiz "mau cua D : "
7          color_C: .asciiz "\nmau cua C : "
8          color_E: .asciiz "\nmau cua E : "
9
10         row_D: .space 100
11         row_C: .space 100
12         row_E: .space 100
13
14         title: .asciiz "\n\n----------------MENU---------------------------------\n"
15         option1: .asciiz "1. Hien thi hinh anh len giao dien console.\n"
16         option2: .asciiz "2. Hien thi hinh anh DCE chi con vien, khong con mau.\n"
17         option3: .asciiz "3. Hien thi hinh anh ECD (duoc hoan doi tu DCE).\n"
18         option4: .asciiz "4. Nhap mau cho D, C, E roi hien thi voi mau vua nhap.\n"
19         option5: .asciiz "5. Thoat.\n"
20         title_end:      .asciiz  "-----------------------------------------------------\n"
21         choose: .asciiz "Lua chon: "
```

Giải thích code:

- Dòng 2 -> 4
    + Gán các dòng biểu diễn chữ D vào trong biến image_D, mỗi dòng ngăn cách nhau bởi ký tự ‘\n’, và ở cuối cùng image_D thêm ký tự ‘?’ để xác định hết ký tự
    + Tương tự như vậy với C và E được lưu vào trong image_C và image_E
- Dòng 6 -> 8: Các title cần hiển thị để cho người dùng biết nên nhập gì
- Dòng 10 -> 12: Khai báo các biến row_D, row_C, row_E với không gian 100 byte
- Dòng 14 -> 21: Tạo cấu trúc menu để dễ dàng kiểm tra các yêu cầu

```
25 # ------------------------------------------------------------------------
26 # s0 : lua chon tu menu
27 # s1 = address(image_D)
28 # s2 = address(image_C)
29 # s3 = address(image_E)
30 # s4 = address(row_D)
31 # s5 = address(row_C)
32 # s6 = address(row_E)
33 # s7 : mau cua D
34 # t8 : mau cua C
35 # t9 : mau cua E
36
37 # D: 22 ky tu ; C: 20 ky tu; E : 16 ky tu
38 # ------------------------------------------------------------------------
```

Giải thích code:

- Comment lại những thanh ghi được sử dụng với chức năng không thay đổi suốt trong một đoạn code lớn hay toàn bộ code

```
40 .text
41
42 # tao menu de de dang test chuong trinh
43 menu:
44         li $v0, 4
45         la $a0, title
46         syscall
47
48         li $v0, 4
49         la $a0, option1
50         syscall
51
52         li $v0, 4
53         la $a0, option2
54         syscall
55
56         li $v0, 4
57         la $a0, option3
58         syscall
59
60         li $v0, 4
61         la $a0, option4
62         syscall
63
64         li $v0, 4
65         la $a0, option5
66         syscall
67
68         li $v0, 4
69         la $a0, title_end
70         syscall
```

Giải thích code:

- Tạo menu bằng cách hiển thị những chuỗi ký tự đã được khởi tạo sẵn
- Sử dụng chức năng syscall với v0 =4 thì in chuỗi trong địa chỉ a0

```
72  # gan mot so bien co ban trong chuong trinh
73  prepare_var:
74          la $s1, image_D         # s1 = address(image_D)
75          la $s4, row_D           # s4 = address(row_D)
76
77          la $s2, image_C         # s1 = address(image_C)
78          la $s5, row_C           # s5 = address(row_C)
79
80          la $s3, image_E         # s1 = address(image_E)
81          la $s6, row_E           # s6 = address(row_E)
82
```

Giải thích code:

- Gán các địa chỉ của các biến vào thanh ghi xác định để chuẩn bị cho các bước tiếp theo

```
83  # lua chon trong menu
84  input_menu:
85          li $v0, 4
86          la $a0, choose
87          syscall
88
89          li $v0, 5
90          syscall
91
92          add $s0, $v0, $zero
93
94  # nhay den chuong trinh lua chon yeu cau
95          beq $s0, 1, main_1
96          beq $s0, 2, main_2
97          beq $s0, 3, main_3
98          beq $s0, 4, main_4
99          beq $s0, 5, exit_pro
100         j input_menu
```

Giải thích code:

- Người dùng sẽ nhập một số nguyên để thể hiện yêu cầu muốn kiểm tra
- Với chức năng syscall, v0 = 5 cho phép đọc một số nguyên mà người dùng nhập
- Dòng 95 -> 100: Để xác định lựa chọn của người dùng và tiến hành thực hiện. Nếu lựa chọn không xác định thì sẽ quay lại menu

```
104 # ---------------------------------------------------------------------------
105 # MAIN 1: GACH DAU DONG THU 1
106 # ---------------------------------------------------------------------------
107 main_1:
108
109 # Display D
110 DD_loop_1:
111         lb $t1, 0($s1)          # t1 = ky tu tai dia chi s1
112         addi $s1, $s1, 1        # tang den dia chi ky tu tiep theo
113         beq $t1, '\n', back_row_D_1     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
114         beq $t1, '?', complete  # hoan thanh in chuoi neu gap ky tu ?
115         sb $t1, 0($s4)          # luu ky tu vao bien row_D
116         addi $s4, $s4, 1        # tang den vi tri ky tu tiep theo
117         j DD_loop_1
118 back_row_D_1:
119         addi $s4, $s4, -22      # tro lai vi tri dia chi ky tu dau tien cua bien row D
120         addi $a0, $s4, 0        # a0 = s4
121         jal print_row
```

Giải thích code:

- **main_1:** bắt đầu thực hiện yêu cầu 1
- **DD_loop_1:** Thực hiện lưu từng dòng của image_D vào trong row_D
  - + Lần lượt gán các ký tự của biến image_D vào trong t1
  - + Nếu t1 không phải ký tự ‘\n’ hoặc ‘?’ thì lưu nó vào trong biến row_D
  - + Nếu là ‘\n’ thì nhảy tới **back_row_D_1**
  - + Nếu là ‘?’ thì đã hoàn thành hiển thị, thoát ra khỏi yêu cầu
- **back_row_D_1:** Hiển thị row_D ra màn hình

```
123 # Display C
124 DC_loop_1:
125         lb $t1, 0($s2)          # t1 = ky tu tai dia chi s2
126         addi $s2, $s2, 1        # tang den dia chi ky tu tiep theo
127         beq $t1, '\n', back_row_C_1     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
128         sb $t1, 0($s5)          # luu ky tu vao bien row_C
129         addi $s5, $s5, 1        # tang den vi tri ky tu tiep theo
130         j DC_loop_1
131 back_row_C_1:
132         addi $s5, $s5, -20      # tro lai vi tri dia chi ky tu dau tien cua bien row C
133         addi $a0, $s5, 0        # a0 = s5
134         jal print_row
135
136
137 # Display E
138 DE_loop_1:
139         lb $t1, 0($s3)          # t1 = ky tu tai dia chi s3
140         addi $s3, $s3, 1        # tang den dia chi ky tu tiep theo
141         beq $t1, '\n', back_row_E_1     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
142         sb $t1, 0($s6)          # luu ky tu vao bien row_E
143         addi $s6, $s6, 1        # tang den vi tri ky tu tiep theo
144         j DE_loop_1
145 back_row_E_1:
146         addi $s6, $s6, -16      # tro lai vi tri dia chi ky tu dau tien cua bien row E
147         addi $a0, $s6, 0        # a0 = s6
148         jal print_row
149         nop
150         jal print_endl
151         nop
152         j DD_loop_1
```

Giải thích code:

- Tương tự như D, ta được row_C và row_E. Hiển thị row_C, row_E rồi xuống dòng. Nhảy lại **DD_loop_1** để tiếp tục hiển thị row_D cho đến khi gặp ‘?’

```
155 # ---------------------------------------------------------------------
156 # MAIN 2: GACH DAU DONG THU 2
157 # ---------------------------------------------------------------------
158 main_2:
159
160 # Display D
161 DD_loop_2:
162         lb $t1, 0($s1)          # t1 = ky tu tai dia chi s1
163         addi $s1, $s1, 1        # tang den dia chi ky tu tiep theo
164         beq $t1, '\n', back_row_D_2     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
165         beq $t1, '?', complete  # hoan thanh in chuoi neu gap ky tu ?
166
167         jal check_int_2
168
169         sb $t1, 0($s4)          # luu ky tu vao bien row_D
170         addi $s4, $s4, 1        # tang den vi tri ky tu tiep theo
171         j DD_loop_2
172 back_row_D_2:
173         addi $s4, $s4, -22      # tro lai vi tri dia chi ky tu dau tien cua bien row D
174         addi $a0, $s4, 0        # a0 = s4
175         jal print_row
176
```

Giải thích code:

- **main_2:** bắt đầu thực hiện yêu cầu 2
- **DD_loop_2:** Giống như **DD_loop_1** nhưng trước khi lưu ký tự vào row_D thì thực hiện nhảy đến **check_int_2** để kiểm tra ký tự vừa load vào t1 xem có phải ký tự màu không.

```
177 # Display C
178 DC_loop_2:
179         lb $t1, 0($s2)          # t1 = ky tu tai dia chi s2
180         addi $s2, $s2, 1        # tang den dia chi ky tu tiep theo
181         beq $t1, '\n', back_row_C_2     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
182
183         jal check_int_2
184
185
186         sb $t1, 0($s5)          # luu ky tu vao bien row_C
187         addi $s5, $s5, 1        # tang den vi tri ky tu tiep theo
188         j DC_loop_2
189 back_row_C_2:
190         addi $s5, $s5, -20      # tro lai vi tri dia chi ky tu dau tien cua bien row C
191         addi $a0, $s5, 0        # a0 = s5
192         jal print_row
```

```
195 # Display E
196 DE_loop_2:
197         lb $t1, 0($s3)          # t1 = ky tu tai dia chi s3
198         addi $s3, $s3, 1        # tang den dia chi ky tu tiep theo
199         beq $t1, '\n', back_row_E_2     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
200
201         jal check_int_2
202
203         sb $t1, 0($s6)          # luu ky tu vao bien row_E
204         addi $s6, $s6, 1        # tang den vi tri ky tu tiep theo
205         j DE_loop_2
206 back_row_E_2:
207         addi $s6, $s6, -16      # tro lai vi tri dia chi ky tu dau tien cua bien row E
208         addi $a0, $s6, 0        # a0 = s6
209         jal print_row
210         nop
211         jal print_endl
212         nop
213         j DD_loop_2
```

Giải thích code:

- Tương tự như D, ta được row_C và row_E. Hiển thị row_C, row_E rồi xuống dòng. Nhảy lại **DD_loop_2** để tiếp tục hiển thị row_D cho đến khi gặp '?'

```
# kiem tra xem ky tu vua doc co phai ky tu mau hay khong
check_int_2:
        slti $t2, $t1, ':'      # t2 =1 neu t1 < '9'
        li $t4, '/'
        slt $t3, $t4, $t1       # t3 =1 neu t1 > '0'
        add $t4, $t2, $t3
        beq $t4, 2, convert_space_2
        jr $ra

# thay mau = ky tu space
convert_space_2:
        li $t1, ' '
        jr $ra
```

Giải thích code:

- **check_int_2:** Kiểm tra xem t1 có phải ký tự số ('0' đến '9') hay không. Nếu đúng thì thay thế bằng ký tự space (' ').

```
# ---------------------------------------------------------------------------
# MAIN 3: GACH DAU DONG THU 3
# ---------------------------------------------------------------------------
main_3:
# Display E
DE_loop_3:
        lb $t1, 0($s3)          # t1 = ky tu tai dia chi s3
        addi $s3, $s3, 1        # tang den dia chi ky tu tiep theo
        beq $t1, '\n', back_row_E_3     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
        beq $t1, '?', complete  # hoan thanh in chuoi neu gap ky tu ?
        sb $t1, 0($s6)          # luu ky tu vao bien row_E
        addi $s6, $s6, 1        # tang den vi tri ky tu tiep theo
        j DE_loop_3
back_row_E_3:
        addi $s6, $s6, -16      # tro lai vi tri dia chi ky tu dau tien cua bien row E
        addi $a0, $s6, 0        # a0 = s6
        jal print_row

# Display C
DC_loop_3:
        lb $t1, 0($s2)          # t1 = ky tu tai dia chi s2
        addi $s2, $s2, 1        # tang den dia chi ky tu tiep theo
        beq $t1, '\n', back_row_C_3     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
        sb $t1, 0($s5)          # luu ky tu vao bien row_C
        addi $s5, $s5, 1        # tang den vi tri ky tu tiep theo
        j DC_loop_3
back_row_C_3:
        addi $s5, $s5, -20      # tro lai vi tri dia chi ky tu dau tien cua bien row C
        addi $a0, $s5, 0        # a0 = s5
        jal print_row
```

```
# Display D
DD_loop_3:
        lb $t1, 0($s1)          # t1 = ky tu tai dia chi s1
        addi $s1, $s1, 1        # tang den dia chi ky tu tiep theo
        beq $t1, '\n', back_row_D_3     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
        sb $t1, 0($s4)          # luu ky tu vao bien row_D
        addi $s4, $s4, 1        # tang den vi tri ky tu tiep theo
        j DD_loop_3
back_row_D_3:
        addi $s4, $s4, -22      # tro lai vi tri dia chi ky tu dau tien cua bien row D
        addi $a0, $s4, 0        # a0 = s4
        jal print_row
        nop
        jal print_endl
        nop
        j DE_loop_3
```

Giải thích code:

- **main_3:** bắt đầu thực hiện yêu cầu 3.
- **main_3** về cấu trúc thì giống như **main_1,** nhưng thay vì hiển thị theo các ký tự theo thứ tự D, C, E như **main_1** thì **main_3** lại hiển thị là E, C, D

```
# ---------------------------------------------------------------------------
# MAIN 4: GACH DAU DONG THU 4
# ---------------------------------------------------------------------------
main_4:

# mau cua D
        # title
color_D_4:
        li $v0, 4
        la $a0, color_D
        syscall

        nop
        nop
        nop

        # doc mau tu ban phim
        li $v0, 12
        syscall
        #kiem tra mau vua nhap co hop le hay khong
        li $t6, 0
        jal check_color_4

        add $s7, $v0, $zero
```

```
# mau cua C
        # title
color_C_4:
        li $v0, 4
        la $a0, color_C
        syscall

        nop
        nop
        nop

        # doc mau tu ban phim
        li $v0, 12
        syscall
        #kiem tra mau vua nhap co hop le hay khong
        li $t6, 1
        jal check_color_4

        add $t8, $v0, $zero
```

```
# mau cua E
       # title
color_E_4:
       li $v0, 4
       la $a0, color_E
       syscall

       nop
       nop
       nop

       # doc mau tu ban phim
       li $v0, 12
       syscall
       #kiem tra mau vua nhap co hop le hay khong
       li $t6, 2
       jal check_color_4
       add $t9, $v0, $zero

       # xuong dong
       li $v0, 12
       li $a0, '\n'
       syscall

# sau khi doc xong tu ban phim thi bat dau xu ly
```

Giải thích code:

- **main_4:** Bắt đầu thực hiện yêu cầu 4.
- Dùng chức năng syscall với v0 = 4 để in chuỗi thông báo người dùng nên nhập màu vào cho ký tự D, C, E
- Dùng chức năng syscall với v0 = 12 để đọc ký tự người dùng nhập từ bàn phím. Đây là màu của từng ký tự.
- Sau đó nhảy đến **check_color_4** để kiểm tra xem ký tự màu có hợp lệ hay không.

```
354 # Display D
355 DD_loop_4:
356         lb $t1, 0($s1)          # t1 = ky tu tai dia chi s1
357         addi $s1, $s1, 1        # tang den dia chi ky tu tiep theo
358         beq $t1, '\n', back_row_D_4     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
359         beq $t1, '?', complete  # hoan thanh in chuoi neu gap ky tu ?
360 
361         jal check_int_4
362 
363         sb $t1, 0($s4)          # luu ky tu vao bien row_D
364         addi $s4, $s4, 1        # tang den vi tri ky tu tiep theo
365         j DD_loop_4
366 back_row_D_4:
367         addi $s4, $s4, -22      # tro lai vi tri dia chi ky tu dau tien cua bien row D
368         addi $a0, $s4, 0        # a0 = s4
369         jal print_row
```

```
371 # Display C
372 DC_loop_4:
373         lb $t1, 0($s2)          # t1 = ky tu tai dia chi s2
374         addi $s2, $s2, 1        # tang den dia chi ky tu tiep theo
375         beq $t1, '\n', back_row_C_4     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
376 
377         jal check_int_4
378 
379 
380         sb $t1, 0($s5)          # luu ky tu vao bien row_C
381         addi $s5, $s5, 1        # tang den vi tri ky tu tiep theo
382         j DC_loop_4
383 back_row_C_4:
384         addi $s5, $s5, -20      # tro lai vi tri dia chi ky tu dau tien cua bien row C
385         addi $a0, $s5, 0        # a0 = s5
386         jal print_row
```

```
389 # Display E
390 DE_loop_4:
391         lb $t1, 0($s3)          # t1 = ky tu tai dia chi s3
392         addi $s3, $s3, 1        # tang den dia chi ky tu tiep theo
393         beq $t1, '\n', back_row_E_4     # neu la ky tu ket thuc dong thi in chuoi ra man hinh
394 
395         jal check_int_4
396 
397         sb $t1, 0($s6)          # luu ky tu vao bien row_E
398         addi $s6, $s6, 1        # tang den vi tri ky tu tiep theo
399         jal DE_loop_4
400 back_row_E_4:
401         addi $s6, $s6, -16      # tro lai vi tri dia chi ky tu dau tien cua bien row E
402         addi $a0, $s6, 0        # a0 = s6
403         jal print_row
404         nop
405         jal print_endl
406         nop
407         j DD_loop_4
```

Giải thích code:

- **DD_loop_4:** Giống như **DD_loop_2,** có thêm lệnh nhảy đến hàm **check_int_4** để kiểm tra xem có phải ký tự màu hay không. Nếu phải thì tiến hành thay thế màu với màu người dùng nhập vào
- **DC_loop_4, DE_loop_4** tương tự như **DD_loop_4**

```
409  # kiem tra xem ky tu vua doc co phai ky tu mau khong
410  check_int_4:
411          slti $t2, $t1, ':'      # t2 =1 neu t1 < '9'
412          li $t4, '/'
413          slt $t3, $t4, $t1       # t3 =1 neu t1 > '0'
414          add $t4, $t2, $t3
415          beq $t4, 2, convert_color_4
416          jr $ra
```

Giải thích code:

- **Check_int_4:** Nếu là ký tự màu (từ ‘0’ đến ‘9’) thì tiến hành đổi màu

```
418  # kiem tra xem ky tu vua doc tu ban phim co phai ky tu mau khong
419  check_color_4:
420          slti $t2, $v0, ':'      # t2 =1 neu v0 <= '9'
421          li $t4, '/'
422          slt $t3, $t4, $v0       # t3 =1 neu v0 >= '0'
423          add $t4, $t2, $t3
424          bne $t4, 2, error_color
425          jr $ra
```

```
427  # Loi mau ky tu
428  error_color:
429          li $v0, 4
430          la $a0, error
431          syscall
432          nop
433          nop
434          nop
435          beq $t6, 0, color_D_4
436          beq $t6, 1, color_C_4
437          beq $t6, 2, color_E_4
```

Giải thích code:

- **check_color_4:** Dùng để kiểm tra màu người dùng nhập vào có hợp lệ hay không
- Nếu là ký tự số (từ ‘0’ đến ‘9’) thì tiếp tục, nếu không phải thì bắt người dùng nhập lại màu bằng hàm **error_color**

```
440  # Xac dinh chu can chuyen mau
441  convert_color_4:
442          beq $t1, '2', convert_color_D_4
443          beq $t1, '1', convert_color_C_4
444          beq $t1, '3', convert_color_E_4
```

Giải thích code:

- **Convert_color_4:** Dùng t1 để xác định màu cần thay là cho ký tự nào.
- Sau đó nhảy đến các hàm thay màu bên tương ứng bên dưới

```
447 # chuyen mau cho D
448 convert_color_D_4:
449         add $t1, $s7, $zero
450         jr $ra
451
452 # chuyen mau cho C
453 convert_color_C_4:
454         add $t1, $t8, $zero
455         jr $ra
456
457 # chuyen mau cho E
458 convert_color_E_4:
459         add $t1, $t9, $zero
460         jr $ra
```

Giải thích code:

- **Convert_color_D_4:** chuyển màu cho ký tự D
- **Convert_color_C_4:** chuyển màu cho ký tự C
- **Convert_color_E_4:** chuyển màu cho ký tự E

```
462 # ------------------------------------------------------------------------
463 # HAM DUNG CHUNG
464 # ------------------------------------------------------------------------
465
466 # xuong dong
467 print_endl:
468         li $v0, 11
469         li $a0, '\n'
470         syscall
471
472         nop
473         jr $ra
474
475 # in dong hien tai
476 print_row:
477         li $v0, 4
478         syscall
479
480         nop
481         jr $ra
482
483 # hoan thanh yeu cau
484 complete:
485         j menu
486
487 # thoat chuong trinh
488 exit_pro:
489         li $v0, 10
490         syscall
```

Giải thích code:

- Đây là các hàm dùng chung trong code của cả 4 yêu cầu
  - **print_endl:** Dùng để xuống dòng
  - **print_row:** Dùng để in dòng hiện tại
  - **complete:** Khi hoàn thành yêu cầu thì quay lại menu
  - **exit_pro:** Thoát khỏi chương trình

Kết quả:

- **Yêu cầu thứ nhất:**

- **Yêu cầu thứ hai**

- **Yêu cầu thứ ba**

```
---------------MENU---------------------------------
1. Hien thi hinh anh len giao dien console.
2. Hien thi hinh anh DCE chi con vien, khong con mau.
3. Hien thi hinh anh ECD (duoc hoan doi tu DCE).
4. Nhap mau cho D, C, E roi hien thi voi mau vua nhap.
5. Thoat.
----------------------------------------------------
Lua chon: 3
  *************
 *3333333333333*                    *************
 *33333********                     *222222222222222*
 *33333*                            *22222******222222*
 *33333********                     *22222*      *22222*
 *3333333333333*      *************  *22222*       *22222*
 *33333********     **111111*****111* *22222*       *22222*
 *33333*          **1111**       **  *22222*       *22222*
 *33333********   *1111*             *22222*      *222222*
 *3333333333333* *11111*             *22222******222222*
  *************  *11111*             *2222222222222222*
                 *11111*             ***************
                  *1111**                 ---
                   *1111****   *****     / o o \
                    **111111***111*      \   > /
 dce.hust.edu.vn      ***********         -----
```

- **Yêu cầu thứ tư (màu của chữ là 3 số cuối mssv của em)**

```
1. Hien thi hinh anh len giao dien console.
2. Hien thi hinh anh DCE chi con vien, khong con mau.
3. Hien thi hinh anh ECD (duoc hoan doi tu DCE).
4. Nhap mau cho D, C, E roi hien thi voi mau vua nhap.
5. Thoat.
----------------------------------------------------
Lua chon: 4

mau cua D : 0
mau cua C : 2
mau cua E : 9
                                            *************
 **************                            *9999999999999*
 *000000000000000*                          *99999*******
 *00000******000000*                        *99999*
 *00000*      *00000*                       *99999*******
 *00000*       *00000*      *************  *9999999999999*
 *00000*       *00000*    **22222*****222* *99999*******
 *00000*       *00000*  **2222**       **  *99999*
 *00000*      *000000*  *2222*             *99999*******
 *00000******000000*  *22222*             *9999999999999*
 *0000000000000000*   *22222*              *************
 ***************      *22222*
       ---             *2222**
     / o o \            *2222****   *****
     \   > /             **222222***222*
      -----                ***********    dce.hust.edu.vn
```