TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

BÁO CÁO CUỐI KỲ
THỰC HÀNH KIẾN TRÚC MÁY TÍNH 20212

Nhóm 10 – Mã lớp: 130937

Giảng viên hướng dẫn: Thầy Lê Bá Vui

Các thành viên thực hiện:

1. **Vũ Tiến Dũng 20205074 Bài 5**
2. **Lê Trường Lam 20204996 Bài 2**

Hà Nội, Ngày 21, Tháng 7, Năm 2022

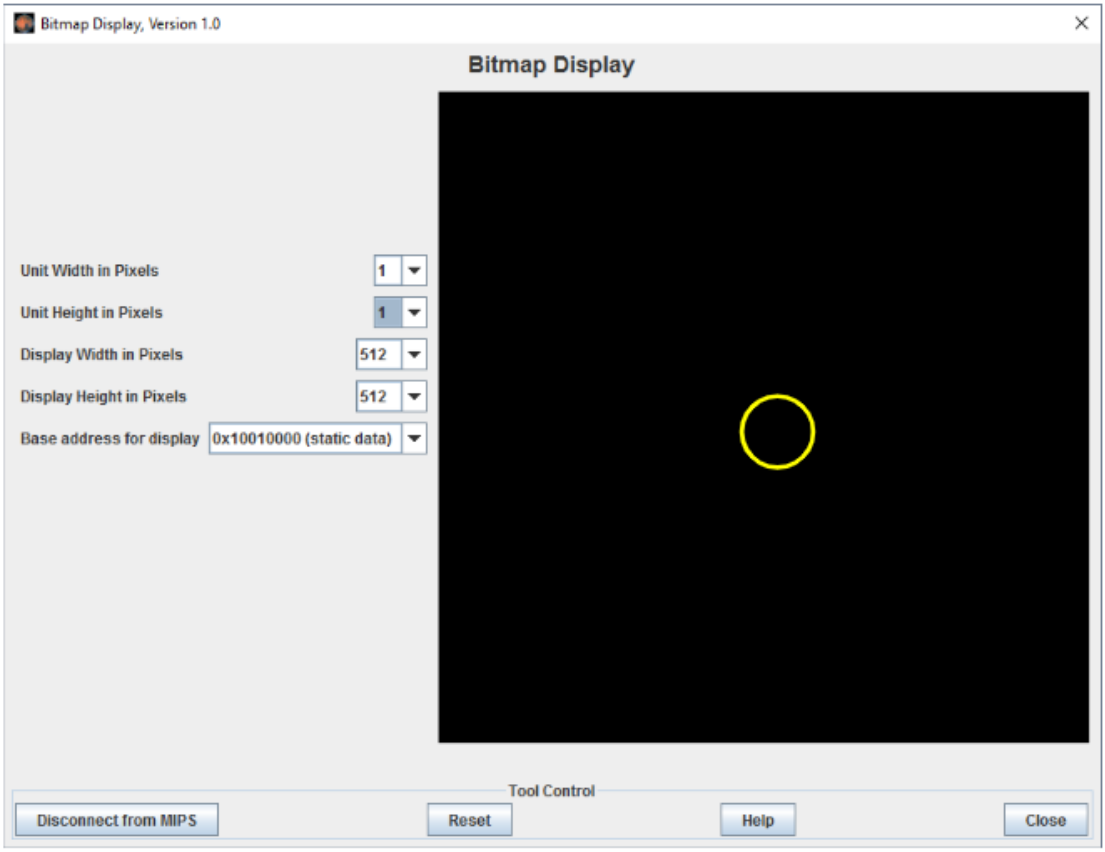
Bài 2:Vẽ hình trên màn hình Bitmap

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 sẽdi 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).
- Vịtrí bóng ban đầu ởgiữa màn hình.

Gợi ý: Đểlàm một đối tượng di chuyển thì chúng ta sẽxóa đối tượng ởvịtrí cũ và vẽđối tượng ởvịtrí mới. Đểxóa đối tượng chúng ta chỉcần vẽđối tượng đó với màu là màu nền.



Bài 5:Biểu thức trung tố hậu tố

Viết chương trình tính giá trị biểu thức bất kỳ bằng phương pháp duyệt biểu thức hậu tố.

Các yêu cầu cụ thể:

1. Nhập vào biểu thức trung tố, ví dụ: 9 + 2 + 8 * 6
2. In ra biểu thức ởdạng hậu tố, ví dụ: 9 2 +8 6* +
3. Tính ra giá trịcủa biểu thức vừa nhập
4. Các hằng số là số nguyên, trong phạm vi từ 0 →99.

Toán tử bao gồm các phép toán cộng, trừ, nhân, chia lấy thương(/), chia lấy dư (%), đóng mở ngoặc.

Thành Viên: Vũ Tiến Dũng – Bài 5:

Mã Nguồn:

```
.data
messenger_1:        .asciiz         "Nhap bieu thuc can tinh: "
messenger_2:        .asciiz         "\nBieu Thuc Hau To La: "
messenger_3:        .asciiz         "\nKet qua sau cua bieu thuc hau to la: "
messenger_4:        .asciiz         "Bieu thuc khong hop le!  Hay nhap lai bieu thuc!"
messenger_5:        .asciiz         "\nBieu Thuc Trung To La: "
A:                                  .space          1000
.text


data:
addi    $t0, $zero, -43             # Phep Cong
addi    $t1, $zero, -45             # Phep Tru
addi    $t2, $zero, -42             # Phep Nhan
```

```
addi    $t3, $zero, -47         # Phep Chia
addi    $t4, $zero, -37         # Phep Chia Lay Du
addi    $t5, $zero, -40         # Ngoac Mo
addi    $t6, $zero, -41         # Ngoac Dong
addi    $s3, $zero, -32         # Dau Cach
addi    $s4, $zero, -10         # Enter, xuong dong
addi    $s2, $zero, 10
addi    $fp, $zero, 100         # Coi 100 la bieu dien cua so 0 trong Stack1

Nhap_bieu_thuc:                 # Ham nhap bieu thuc
li          $v0, 54
la          $a0, messenger_1
la          $a1, A
la          $a2, 1000           # Toi da 1000 ky tu
syscall

addi        $s0, $gp, 0         # Stack 1
la          $t7, A              # Luu dia chi cua bieu thuc vua nhap vao $t7
Xet_Thanh_Phan:
Read_Char:
        lb          $t8, 0($t7)
        beq         $t8, 0, Kiem_Tra_Hop_Le_2       # Kiem tra xem gap ky tu NULL chua
        beq         $t8, '\n', Kiem_Tra_Hop_Le_2    # Kiem tra xem co phai Enter khong
        beq         $t8, '+', push_s1               # Phep Cong
        beq         $t8, '-', push_s1               # Phep Tru
        beq         $t8, '*', push_s1               # Phep Nhan
        beq         $t8, '/', push_s1               # Phep Chia
        beq         $t8, '%', push_s1               # Chia Lay Du
```

```
        beq     $t8, '(', push_s1       # Ngoac Mo
        beq     $t8, ')', push_s1       # Ngoac Dong
        beq     $t8, ' ', continue_char  # Dau Cach

        slti    $t9, $t8, 48            # Kiem tra xem co phai so hay khong
        bne     $t9, $zero, Nhap_Lai    # Neu khong phai cac phep toan, dau cach,
                                        # Kiem_Tra_Hop_Le_2 hay so thi bieu thuc ko hop le
        slti    $t9, $t8, 58            # 48 <= $t8 < 58
        bne     $t9, $zero, Interger    # Neu la so thi chuyen den Interger de phan tich
        j       Nhap_Lai                # Cac truong hop con lai deu ko hop le

push_s1:    # Luu cac phep toan vao stack_1
        sub     $t8, $0, $t8            # Khi luu vao stack -$t8 se dai dien cho cac toan tu
        sw      $t8, 0($s0)
        addi    $t7, $t7, 1
        addi    $s0, $s0, 4
        j       Read_Char               # Sau khi luu thi xet phan tu tiep

continue_char:                          # Ham nay se xet phan tu tiep theo
        addi    $t7, $t7, 1             # Khi gap dau cach thi bo qua xet phan tu tiep
        j       Read_Char

Nhap_Lai:   # Ham de nhap lai ham khi ham khong dung dinh dang
        li      $v0, 55
        la      $a0, messenger_4
        syscall
        addi    $s0, $gp, 0                 # Reset lại gia tri cua stack1
        Reset:
```

```
        lw      $t8, 0($s0)
        beq     $t8, $zero, Nhap_bieu_thuc  # Khi da reset xong thi nhap lai
        sw      $zero, 0($s0)               # Reset gia tri cua stack1
        addi    $s0, $s0, 4
        j       Reset

Store_Digit:    # Neu day la so co 1 chu so thi luu truc tiep vao stack_1
        beq     $s5, 0, Store_Zero          # Xet day co la so 0 hay khong
        sw      $s5, 0($s0)                 # Neu khac 0 thi luu vao stack1
        addi    $t7, $t7, 1
        addi    $s0, $s0, 4
        j       Read_Char

Store_Zero:     # Ham nay de luu 100 thay cho so 0 vao stack1
        addi    $s5, $zero, 100
        sw      $s5, 0($s0)
        addi    $t7, $t7, 1
        addi    $s0, $s0, 4
        j       Read_Char

Interger_2:     # Ham se xet so co 2 chu so se luu vao stack_1
        multu   $s5, $s2                    # Lay so hang chuc $s2 = 10
        mflo    $s5
        addi    $t8, $t8, -48               # Tru ma ascii cua hang don vi de lay Integer tuong ung
        add     $s5, $s5, $t8               # Cong vao de lay dc so can tim
        sw      $s5, 0($s0)                 # Luu so 2 chu so vao tack1
        addi    $s0, $s0, 4
        addi    $t7, $t7, 2                 # Xet phan tu o vi tri i+2 de xet day co phai so 3 chu so ko
```

```
        lb          $t8, 0($t7)
        beq         $t8, 0, Kiem_Tra_Hop_Le_2 # Neu xet xong bieu thuc dau vao thi chuyen
        beq         $t8, '\n', Kiem_Tra_Hop_Le_2
        slti        $t9, $t8, 58                # Neu la so co 3 chu so thi nhap lai bieu thuc
        beq         $t9, $zero, Nhap_Lai
        slti        $t9, $t8, 48                #     48 <= 48 <58 ?
        beq         $t9, $zero, Nhap_Lai
        j           Read_Char

Interger:       # Ham xet so co 1 chu so
        addi    $s5, $t8, -48           # Ta lay ascii ta tru cho 48 se ra so Integer tuong ung
        lb      $t8, 1($t7)             # Xet ky tu tiep theo xem co phai la so 2 chu so hay khong
        slti    $t9, $t8, 48
        bne     $t9, $zero, Store_Digit         # Neu la so co 1 chu so thi t se luu luon vao Stack_1
        slti    $t9, $t8, 58
        bne     $t9, $zero, Interger_2  # Neu la so co 2 chu so thi t se sang de chuyen thanh so 2
                                        # chu so
        beq     $t8, 0, Kiem_Tra_Hop_Le_2 # Kiem tra xem co phai ky tu 'NULL' hay khong
        beq     $t8, $s4, Kiem_Tra_Hop_Le_2     # Kiem tra xem co phai ky tu '\n' hay khong

# Tat ca da duoc luu trong Stack_1
# Giai phong thanh ghi $t7, $t8, $t9, $s2, $s3, $s4, $s5, $s6, $s7
# $s0 la stack_1 luu cac toan tu, toan hang cua bieu thuc
Kiem_Tra_Hop_Le_2:      # Kiem tra xem Bieu Thuc Nhap co Hop Le hay Khong
        addi    $s0, $gp, 0
        addi    $fp, $zero, 0

        lw      $t8, 0($s0)     # Xet Phan tu dau tien Neu la cac toan tu thi BT ko hop le
```

```
        beq     $t8, $t2, Nhap_Lai      # Phep Nhan
        beq     $t8, $t3, Nhap_Lai      # Phep Chia
        beq     $t8, $t4, Nhap_Lai      # Chia Lay Du
        beq     $t8, $t6, Nhap_Lai      # Ngoac Dong

Loop:
        lw      $t8, 0($s0)
        beq     $t8, $t0, Cong          # Phep Cong
        beq     $t8, $t1, Cong          # Phep Cong
        beq     $t8, $t2, Cong          # Phep Nhan
        beq     $t8, $t3, Cong          # Phep Chia
        beq     $t8, $t4, Cong          # Chia Lay Du
        beq     $t8, $t5, Mo_Ngoac      # Mo Ngoac
        beq     $t8, $t6, Dong_Ngoac    # Dong Ngoac
        beq     $t8, 0, Done            # NULL
        j       Number

        Cong:
                lw      $t8, 4($s0)             # Xet phần tử sau dấu cộng
                beq     $t8, $t0, Nhap_Lai      # Phep Cong
                beq     $t8, $t1, Nhap_Lai      # Phep Tru
                beq     $t8, $t2, Nhap_Lai      # Nhan
                beq     $t8, $t3, Nhap_Lai      # Chia
                beq     $t8, $t4, Nhap_Lai      # Chia Lay Du
                beq     $t8, $t6, Nhap_Lai      # Dong Ngoac
                beq     $t8, 0  , Nhap_Lai      # NULL
                addi    $s0, $s0, 4
                j       Loop
```

```
Mo_Ngoac:                               # Xet Cac phan tu sau dau ngoac mo
        lw      $t8, 4($s0)
        beq     $t8, $t2, Nhap_Lai      # Nhan
        beq     $t8, $t3, Nhap_Lai      # Chia
        beq     $t8, $t4, Nhap_Lai      # Chia Lay Du
        beq     $t8, 0  , Nhap_Lai      # NULL
        addi    $s0, $s0, 4
        addi    $fp, $fp, 1             # Gap ngoac mo thì fp+1
        j       Loop

Dong_Ngoac:                             # Xet Cac phan tu sau dau ngoac dong
        lw      $t8, 4($s0)
        beq     $t8, $t5, Nhap_Lai      # Mo Ngoac
        addi    $fp, $fp, -1            # Gap ngoac dong thì fp -1
        blt     $fp, 0, Nhap_Lai    # Neu dau dong ngoac truoc dau ngoac nhap lai
        beq     $t8, 0, Done
        slti    $t9, $t8, 0             # Neu khong phai toan tu nhap lai
        beq     $t9, 0, Nhap_Lai
        addi    $s0, $s0, 4
        j       Loop

Number:         # Xet phan tu sau cac toan hang
        lw      $t8, 4($s0)
        beq     $t8, $t5, Nhap_Lai      # Mo Ngoac
        beq     $t8, 0  , Done
        addi    $s0, $s0, 4
        j       Loop
```

```
Done:       # Xet cac dau ngoac co dung vi tri hay khong
bne         $fp, 0, Nhap_Lai
j           Print_Infix

Convert_To_Hau_To:
# Tat ca da duoc luu trong Stack_1
# Giai phong thanh ghi $t7, $t8, $t9, $s2, $s3, $s4, $s5, $s6, $s7, $fp
# $s0 la stack_1 luu cac toan tu, toan hang cua bieu thuc

# Chuyen sang hau to
addi    $s0, $gp, 0                 # Stack 1 Chua Cac Toan Tu, Toan Hang Sau khi da check
addi    $s1, $sp, 4                 # Stack 2 Chua Cac Toan Tu De Duyet
addi    $fp, $sp, -4            # Stack 3 Chua PostFix

Head_Sub_Add:
lw          $t7, 0($s0)
beq         $t7, -45, Check_Sub     # Check phep Tru
beq         $t7, -43, Check_Add     # Check phep Cong
Loop_2:             # Xet tung phan tu
    lw          $t7, 0($s0)
    beq         $t7, $t0, Dau_Cong
    beq         $t7, $t1, Dau_Cong
    beq         $t7, $t2, Dau_Nhan
    beq         $t7, $t3, Dau_Nhan
    beq         $t7, $t4, Dau_Nhan
    beq         $t7, $t5, Ngoac_Mo
    beq         $t7, $t6, Ngoac_Dong
```

```
        beq     $t7, 0  , NULL      # Khi xet het phan tu cua bieu thuc chuyen den NUll
        beq     $t7, $s4, NULL      # de dua cac phan tu con lai cua Stack 2 sang Stack 3
        j       Number_1

Check_Sub:      # Kiem tra phan tu dau co la dau -
        sw      $zero, 0($fp)       # (-3) = 03-
        lw      $t8, 4($s0)
        sw      $t8, -4($fp)
        sw      $t7, -8($fp)
        addi    $fp, $fp, -12
        addi    $s0, $s0, 8
        j       Loop_2

Check_Add:      # Nau phan tu dau la dau + thi bỏ qua
        addi    $s0, $s0, 4
        j       Loop_2


Ngoac_Mo:       # Khi la dau ngoac mo
        lw      $t7, 4($s0)             # Xet phan tu tiep la gi
        addi    $s0, $s0, 4
        sw      $t5, 0($s1)             # Luu Ngoac Mo vao Stack 2
        addi    $s1, $s1, 4
        beq     $t7, -45, Check_Sub     # Neu la - thi vao Check_Sub
        beq     $t7, -44, Check_Add     # Tuong tu
        j       Loop_2
```

```
Ngoac_Dong:     # Khi dong ngoac
        lw      $t7, -4($s1)            # Xet cac phan tu trong stack chua toan tu
        beq     $t5, $t7, Continues_3 # Neu la ngoac mo thi xet cac phan tu tiep
        sw      $t7, 0($fp)                     # Neu khong phai chuyen cac toan tu
sang Stack 3
        sw      $zero, -4($s1)          # Reset lai phan tu da chuyen cua Stack 2
        addi    $s1, $s1, -4            #
        addi    $fp, $fp, -4
        j       Ngoac_Dong

Continues_3:
        sw      $zero, -4($s1)          # Xoa ngoac mo khoi Stack 2 chua toan tu
        addi    $s1, $s1, -4
        j       Continues_2

Dau_Cong:                               # Neu la cong hoac tru thi
        lw      $t8, -4($s1)
        beq     $t8, 0, Store_2         # Neu Stack 2 trong thi luu truc tiep vao
        beq     $t8, $t5, Store_2       # Neu truoc no la dau ngoac thi luu vao
        j       Store                   # Neu khong phai chuyen sang Store


Store_2:
        sw      $t7, 0($s1)             # Dung de luu cac toan hang vao
Stack 2
        addi    $s1, $s1, 4
        addi    $s0, $s0, 4
        j       Loop_2
```

```
Dau_Nhan:                                   # Dau nhan
        lw      $t8, -4($s1)                # Xet phan tu trong Stack 2
        beq     $t8, $t2, Store_N           # Neu gap dau nhan thi luu chuyen
        beq     $t8, $t3, Store_N           # Neu gap dau chia thi luu chuyen
        beq     $t8, $t4, Store_N           # Neu gap dau chia du thi chuyen

        sw      $t7, 0($s1)                 # Neu khong phai thi luu luon vao Stack2
        addi    $s1, $s1, 4
        addi    $s0, $s0, 4
        j       Loop_2

Store_N:
        sw      $t8, 0($fp)                 # Luu phan tu dau Stack 2 vao Stack 3 Postfix
        addi    $fp, $fp, -4
        addi    $s0, $s0, 4
        sw      $t7, -4($s1)            # Sau do luu phan tu dang xet vao Stack 2
        j       Loop_2

Continues_2:                            # Xet tiep phan tu
        addi    $s0, $s0, 4
        j       Loop_2

NULL:# Sau khi xet het bieu thuc thi dua cac toan tu con lai cua st2->st3
        lw      $t8, -4($s1)
        beq     $t8, 0, Done_2          # Sau khi dua het thi chuyen xong
```

```
        sw      $zero, -4($s1)      # Khi lay phan tu ra khoi st2 thi reset
        addi    $s1, $s1, -4
        sw      $t8, 0($fp)         # Luu vao Stack 3
        addi    $fp, $fp, -4
        j       NULL

Store:  # Khi la dau cong thi
        sw      $t8, 0($fp)         # Chuyen phan tu dau tien cua Stack2 -> Stack3
        addi    $fp, $fp, -4
        addi    $s0, $s0, 4
        lw      $t8, -8($s1)        # Xet tiep phan tu Neu la nhung phan tu co do uu
        beq     $t8, $t0, Store_3   # Uu tien thap ra Stack 3
        beq     $t8, $t1, Store_3
        sw      $t7, -4($s1)        # Neu khong thi luu toan tu vao Stack 2
        j       Loop_2
Store_3:    #
        sw      $t8, 0($fp)
        addi    $fp, $fp, -4
        sw      $zero, -4($s1)
        sw      $t7, -8($s1)
        addi    $s1, $s1, -4
        j       Loop_2

Store_1:
        sw      $t8, 0($fp)
        addi    $fp, $fp, -4
        addi    $s0, $s0, 4
        sw      $t7, -4($s1)
```

```
        j               Loop_2

Number_1:   # Gap toan hang
        sw              $t7, 0($fp)             # Gap toan hang luu luon vao Stack 3
        addi    $fp, $fp, -4
        addi    $s0, $s0, 4
        j               Loop_2


Done_2:
        j               Print_Postfix

Print_number: # Nếu gặp số thì in số
li      $v0, 1
addi    $a0, $t8, 0
syscall
li              $v0, 11
addi    $a0, $zero, 32
syscall
j               X_Loop

Print_operater:# Nếu là toán tử thì in kiểu char
li              $v0, 11
abs             $a0, $t8
syscall
li              $v0, 11
addi    $a0, $zero, 32
syscall
```

```
j               X_Loop

X_Loop:
beq             $t9, 1, Loop_3
j               Loop_4

Print_Infix:    # In hậu tố
addi    $s0, $gp, 0
li              $v0, 4
la              $a0, messenger_5
syscall
addi    $t9, $0, 1
Loop_3: # Lap de in phan tu
        lw              $t8, 0($s0)
        addi    $s0, $s0, 4
        beq             $t8, 100, Print_zero
        beq             $t8, 0, Convert_To_Hau_To
        blt             $t8, 0, Print_operater
        j               Print_number
Print_zero:
        addi    $t8, $0, 0
        j               Print_number


Print_Postfix:  # In ra Trung tố
addi    $fp, $fp, -4
addi    $s0, $sp, -4
li              $v0, 4
```

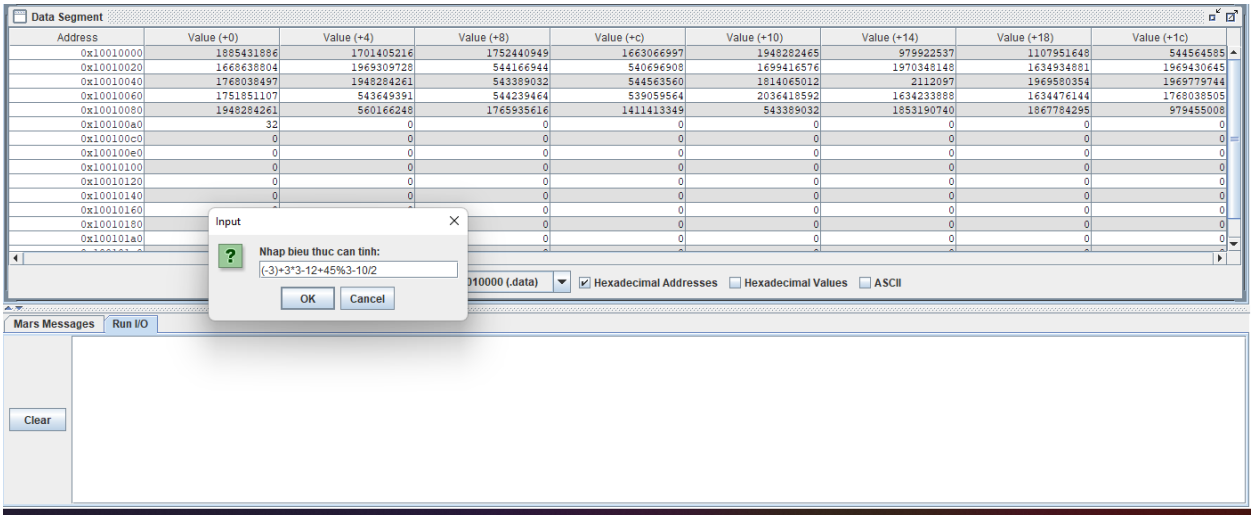
```
la              $a0, messenger_2
syscall
addi    $t9, $0, 0
Loop_4:             # Lap de in phan tu
        lw          $t8, 0($s0)
        addi    $s0, $s0, -4
        beq         $t8, 100, Print_zero
        beq         $s0, $fp, Calculater
        blt         $t8, 0, Print_operater
        j           Print_number

# Tinh Toan Bieu Thuc Hau To

Calculater:
addi    $s2, $0, 0

addi    $fp, $fp, 4             # Lam dieu kien ngat
addi    $s0, $sp, -4    # Stack luu Postfix
addi    $s1, $sp, 20    # Luu cac toan hang
Loop_5:             # Xet cac phan tu
        lw          $t7, 0($s0)
        beq         $s0, $fp, Print_Values
        addi    $s0, $s0, -4
        beq         $t7, $t0, _Cong
        beq         $t7, $t1, _Tru
        beq         $t7, $t2, _Nhan
        beq         $t7, $t3, _Chia
        beq         $t7, $t4, _Chia_Du
```

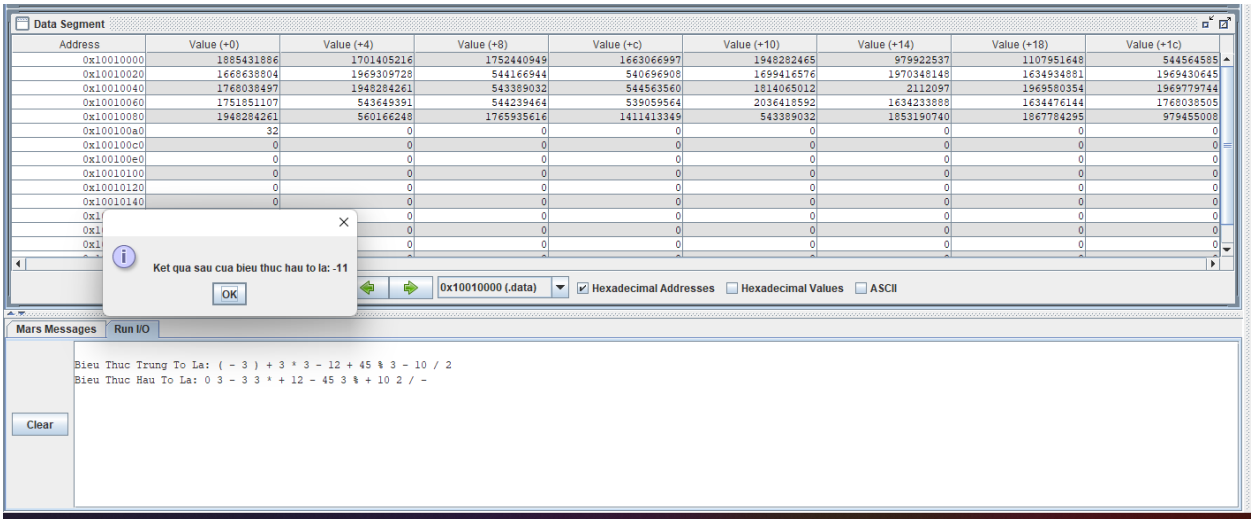
```
        beq             $t7, 100, _Zero
        j               _Number

_Zero:
        addi    $t7, $0, 0
        j               _Number

Print_Values:
addi    $v0, $0, 56
la              $a0, messenger_3
lw              $a1, 20($sp)
syscall

Exit:
addi    $v0, $0, 10
syscall

_Cong:
lw              $s3, -4($s1)
sw              $zero, -4($s1)
lw              $s2, -8($s1)
add             $s2, $s3, $s2
addi    $s1, $s1, -4
sw              $s2, -4($s1)
j               Loop_5
_Tru:
lw              $s3, -4($s1)
sw              $zero, -4($s1)
```

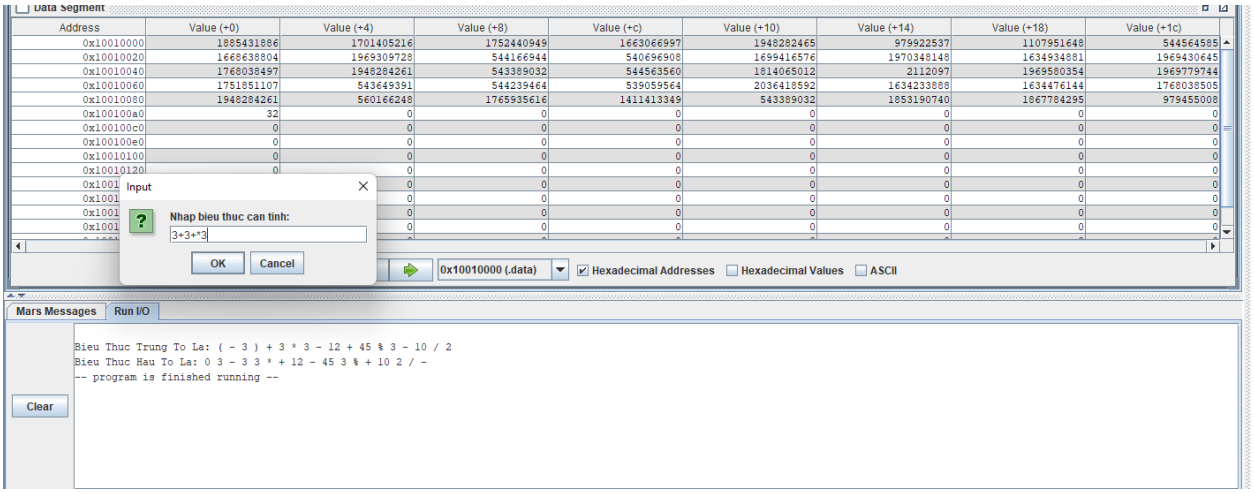
```
lw      $s2, -8($s1)
sub     $s2, $s2, $s3
addi    $s1, $s1, -4
sw      $s2, -4($s1)
j       Loop_5
_Nhan:
lw      $s3, -4($s1)
sw      $zero, -4($s1)
lw      $s2, -8($s1)
mult    $s3, $s2
mflo    $s2
addi    $s1, $s1, -4
sw      $s2, -4($s1)
j       Loop_5
_Chia:
lw      $s3, -4($s1)
sw      $zero, -4($s1)
lw      $s2, -8($s1)
div     $s2, $s2, $s3
mflo    $s2
addi    $s1, $s1, -4
sw      $s2, -4($s1)
j       Loop_5
_Chia_Du:
lw      $s3, -4($s1)
sw      $zero, -4($s1)
lw      $s2, -8($s1)
div     $s2, $s2, $s3
```

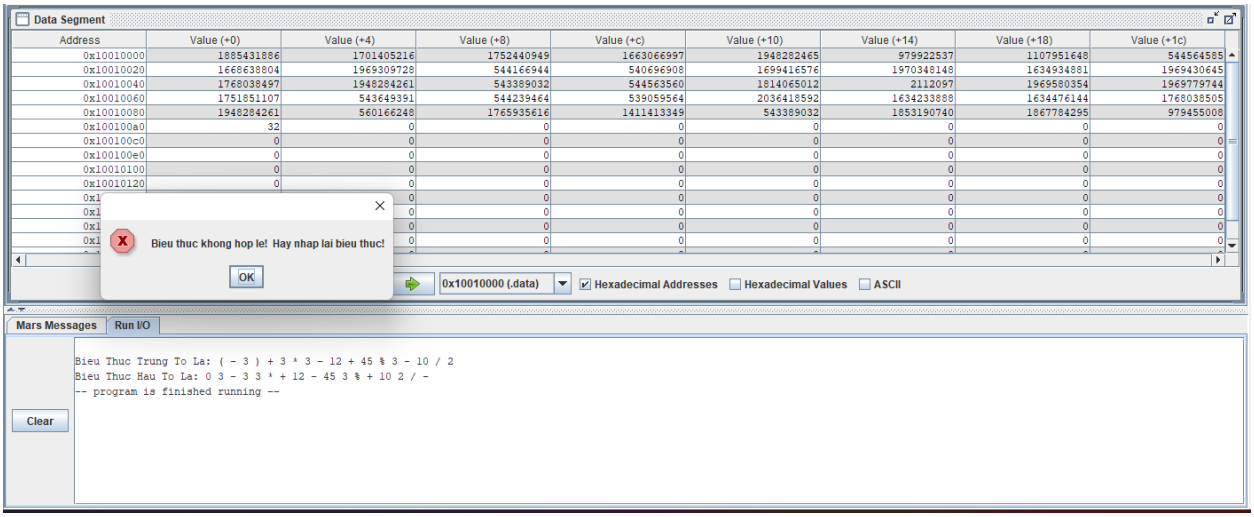
```
mfhi    $s2
addi    $s1, $s1, -4
sw      $s2, -4($s1)
j       Loop_5
_Number:
sw      $t7, 0($s1)
addi    $s1, $s1, 4
j       Loop_5
```

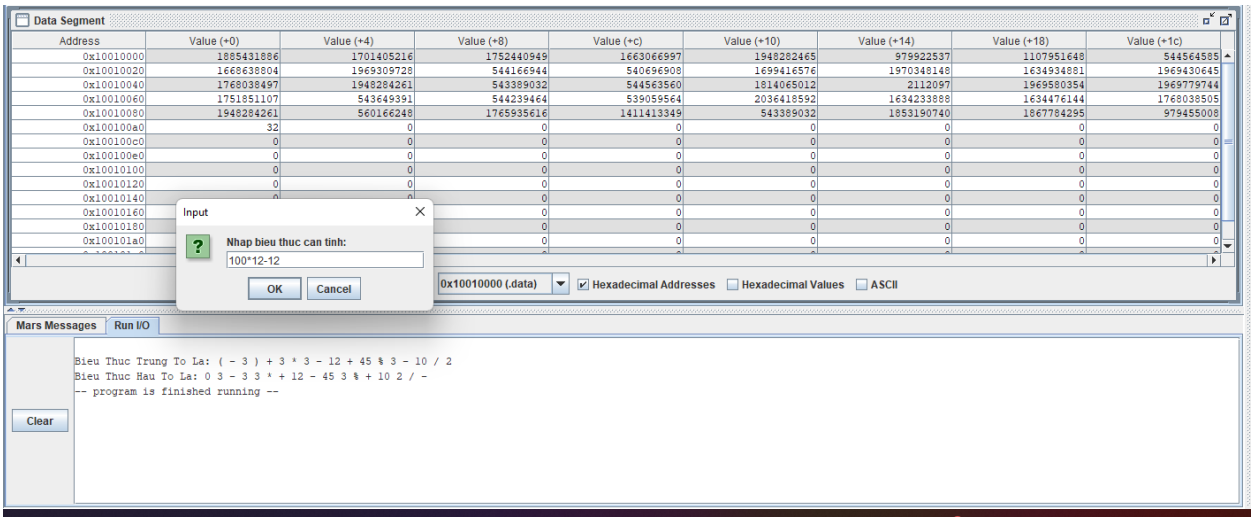
Kết Quả:

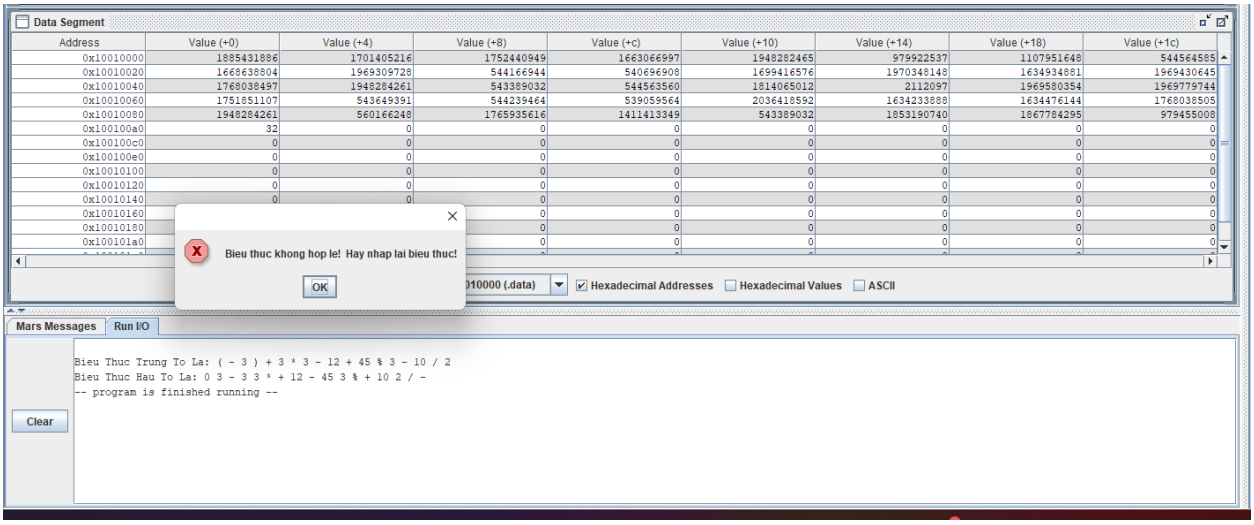





Data Segment
Input
Nhap bieu thuc can tinh:
3+3+*3
OK
Cancel
0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII
Mars Messages
Run I/O
Bieu Thuc Trung To La: (- 3) + 3 * 3 - 12 + 45 % 3 - 10 / 2
Bieu Thuc Hau To La: 0 3 - 3 3 * + 12 - 45 3 % + 10 2 / -
-- program is finished running --
Clear


Data Segment
Bieu thuc khong hop le! Hay nhap lai bieu thuc!
OK
0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII
Mars Messages
Run I/O
Bieu Thuc Trung To La: (- 3) + 3 * 3 - 12 + 45 % 3 - 10 / 2
Bieu Thuc Hau To La: 0 3 - 3 3 * + 12 - 45 3 % + 10 2 / -
-- program is finished running --
Clear


Data Segment
Input
Nhap bieu thuc can tinh:
100*12-12
OK
Cancel
0x10010000 (.data)
Hexadecimal Addresses
Hexadecimal Values
ASCII
Mars Messages
Run I/O
Bieu Thuc Trung To La: (- 3) + 3 * 3 - 12 + 45 % 3 - 10 / 2
Bieu Thuc Hau To La: 0 3 - 3 3 * + 12 - 45 3 % + 10 2 / -
-- program is finished running --
Clear



Giải Thích Cách làm và mã nguồn

Bài làm của e chia thành 5 phần:

Phần 1: Sau khi nhập biểu thức ta kiểm tra các điều kiện đầu vào các các số từ 0 -> 99, các phép toán là ‘+’, ‘-‘,’*’, ‘/’, ‘%’, dấu ngoặc mở, dấu ngoặc đóng.

Khi không đáp ứng điều kiện sẽ nhập lại biểu thức

Phần 2: Xét biểu thức có đúng định dạng không, ví dụ như sau dấu đóng ngoặc phải là các toán tử, không phải là các số, sâu cac phép toán như ‘*’, ‘/’, ‘%’ phải là các toán hạng hoặc dấu ngoặc ‘(‘ , … Nếu không đáp ứng điều kiện sẽ phải nhập lại biểu thức. Sau khi xét xong thì in biểu thức trung tố ra màn hình

Phần 3: Sau khi đã duyệt xong biểu thức, thì ta đến bước chuyển sang hậu tố, ta sẽ tạo ra 2 stack_1, 1 stack lưu các phần từ sau Postfix, 1 stack_2 dùng để lưu các toán tử. Ý tưởng chuyển e được học trong môn Cấu trúc dữ liệu và giải thuật:

Khi gặp các số thì đưa trực tiếp vào Stack_1

Khi gặp các toán tử nếu Stack_2 trống thì đưa trực tiếp vào stack_2

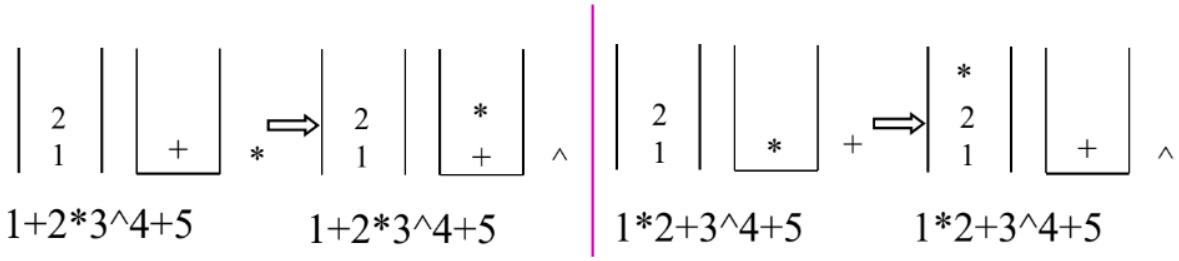
Chuyển biểu thức dạng trung tố về dạng hậu tố (Infix to Postfix Conversion)

- Chuyển đổi biểu thức dạng trung tố về dạng hậu tố để có thể tính được dễ dàng.
- Ta xét cách chuyển đổi biểu thức trung tố với các phép toán ***cộng, trừ, nhân, chia, luỹ thừa và các dấu ngoặc*** về dạng hậu tố.
- Trước hết nhắc lại qui tắc tính giá trị biểu thức trung tố như vậy:
  - ***Thứ tự ưu tiên***: Luỹ thừa; Nhân/Chia; Cộng/Trừ
  - ***Qui tắc kết hợp***: Cho biết khi hai phép toán có cùng thứ tự ưu tiên thì cần thực hiện phép toán nào trước.
    - Luỹ thừa: Phải qua trái. Ví dụ: 2^2^3= 2^(2^3)=256
    - Nhân/Chia: Trái qua phải.
    - Cộng/Trừ: Trái qua phải
  - ***Dấu ngoặc*** được ưu tiên hơn cả thứ tự ưu tiên và qui tắc kết hợp

- Nếu phép toán có tính kết hợp trái (left associative), thì phép toán ở đỉnh ngăn xếp cần đưa ra

```
4-4-4 => 4 4 - 4 -
```

  - **Qui tắc cơ bản**: Nếu phép toán đang xét có thứ tự ưu tiên ***thấp hơn*** so với phép toán ở đầu ngăn xếp, thì phép toán ở đầu ngăn xếp phải dời ngăn xếp.



- Dấu mở ngoặc có thứ tự ưu tiên hơn là phép toán khi nó được xét như là *ký tự đầu vào (input symbol)* (nghĩa là không có gì dời khỏi ngăn xếp). Dấu mở ngoặc có thứ tự ưu tiên thấp hơn phép toán khi nó *ở ngăn xếp*.
- Dấu đóng ngoặc sẽ đẩy phép toán ra khỏi ngăn xếp cho đến khi gặp dấu mở ngoặc dời khỏi ngăn xếp. Các phép toán sẽ được ghi ra còn các dấu ngoặc thì không được ghi ra.

- Duyệt biểu thức từ trái qua phải:
- Nếu gặp *toán hạng (Operands)*: đưa ra tức thì.
- Nếu gặp *dấu mở ngoặc* thì nạp nó vào ngăn xếp.
- Nếu gặp *dấu đóng ngoặc*: Đẩy ký hiệu ra khỏi ngăn xếp cho đến khi gặp dấu mở ngoặc đầu tiên được đẩy ra.
- Nếu gặp *phép toán (Operator)*: Đưa ra khỏi ngăn xếp tất cả các phép toán cho đến khi gặp phép toán có thứ tự ưu tiên thấp hơn hoặc gặp phép toán có tính kết hợp phải và có cùng thứ tự ưu tiên. Sau đó nạp phép toán đang xét vào ngăn xếp.
- *Khi duyệt hết biểu thức*: Đưa tất cả các phép toán còn lại ra khỏi ngăn xếp.

Phần 4: Khi chuyển sang biểu thức hậu tố thì ta sẽ đến bước tính toán từ biểu thức hậu tố

Từ Stack chưa biểu thức hậu tố vừa chuyển ta lấy lần lượt từng phần tự.

Tạo Stack_2 để chứa số Stack_3 để chưa các toán tử và dấu mở, đóng ngoặc

Khi gặp phép toán thì từ Stack_2 ta lấy 2 phần tử ở đầu để thực hiện phép toán sau khi thực hiện phép toán thì giảm Tack_2 đi 1 ô và chèn kết quả vừa nhập vào đầu Stack_2;

Khi duyệt xong biểu thức hậu tố thì Ta in ra một hộp thông báo ghi kết quả vừa tính được.

Phần 5: là phần in ra các biểu thức trung tố và hậu tố ở box dưới của phần mềm. Sau khi In kết quả tính toán thì ta kết thúc chương trình.

Thành viên thực hiện Lê Trường Lam

Bài 2:

CODE:

```
.eqv KEY_CODE 0xFFFF0004  # ASCII code to show, 1 byte
.data
                L :      .asciiz "a"
                R : .asciiz "d"
                U:       .asciiz "w"
                D:       .asciiz "s"
.text
                li $k0, KEY_CODE    # chua ký tu nhap vao
                #Ve bong tai tam man hinh
                #Tam cua bong la (x0,y0)
                #a0 = x0
                #a1 = y0
                #a2 = color
                #a3 = radius
.eqv    YELLOW 0x00FFFF00
.eqv    MONITOR_SCREEN 0x10010000
.text
                li $v1, MONITOR_SCREEN
#Gia tri ban dau tai tam  # (x,y) = (256,256)
                li                $a0, 256
```

```
            li      $a1, 256
            li      $a3, 20          #  Ban kinh cua qua bong R =
20
            li      $a2, YELLOW    # Bong mau vang
            li      $s1,0               # dem = 0
            addi   $s7, $0, 512        #luu do rong vao thanh ghi s7
            jal     DrawCircle   # Ve hinh tron
            nop
moving:         #Thuc hien di chuyen bong
            beq   $t0,97,left          #$t0 = 'a'
            beq   $t0,100,right        #$t0 = 'd'
            beq   $t0,115,down         #$t0 = 's'
            beq   $t0,119,up           #$t0 = 'w'
            beq   $t0,120,speedup   #$to = 'x' nhanh
            beq   $t0,122,speeddown   #$to = 'z' cham
            j       Input
        speedup:
            addi   $s1,$s1,1
            beq   $t9,97,leftnext2          #$t9 = 'a'
            beq   $t9,100,rightnext2        #$t9 = 'd'
            beq   $t9,115,downnext2             #$t9 = 's'
            beq   $t9,119,upnext2               #$t9 = 'w'
            j Input
        speeddown:
            addi   $s1,$s1,-1
            beq   $t9,97,leftnext1          #$t9 = 'a'
            beq   $t9,100,rightnext1        #$t9 = 'd'
            beq   $t9,115,downnext1             #$t9 = 's'
            beq   $t9,119,upnext1               #$t9 = 'w'
            j       Input
        leftnext2:
            j       left2
        rightnext2:
            j       right2
        downnext2:
            j       down2
```

```
upnext2      :
        j           up2
leftnext1:
        j           left1
rightnext1:
        j           right1
downnext1:
        j           down1
upnext1      :
        j           up1
left:   #Thuc hien di chuyen sang trai
        move $t9,$t0
        bgtz   $s1,left2
        bltz   $s1,left1
        li          $a2,0x00000000    #color = black
        jal         DrawCircle     # to lai mau nen
        addi   $a0,$a0,-1          #x0 = x0 - 1
        add    $a1,$a1, $0         #y0 = y0
        li          $a2, YELLOW              #color = yellow
        jal     DrawCircle
        jal     Pause
        bltu   $a0,20,reboundRight      #Neu x0 = 20 tuc den thanh trai
thi thu hien bat phai
        j           Input
left1:  #Thuc hien di chuyen sang trai
        li          $a2,0x00000000    #color = black
        jal     DrawCircle     # to lai mau nen
        addi   $a0,$a0,-1    #x0 = x0 - 1
        add    $a1,$a1, $0         #y0 = y0
        li           $a2, YELLOW              #color = yellow
        jal     DrawCircle
        jal     Pause1
        bltu   $a0,20,reboundRight      #Neu x0 = 20 tuc den thanh trai
thi thu hien bat phai
        j           Input
left2:  #Thuc hien di chuyen sang trai
```

```
            li      $a2,0x00000000    #color = black
            jal     DrawCircle     # to lai mau nen
            addi    $a0,$a0,-1    #x0 = x0 - 1
            add     $a1,$a1, $0         #y0 = y0
            li      $a2, YELLOW               #color = yellow
            jal     DrawCircle
            jal     Pause2
            bltu    $a0,20,reboundRight       #Neu x0 = 20 tuc den thanh trai
thi thu hien bat phai
            j       Input
        right:  #Thuc hien di chuyen sang phai
            move $t9,$t0
            bgtz    $s1,right2
            bltz    $s1,right1
            li      $a2,0x00000000    #color = black
            jal     DrawCircle
            addi    $a0,$a0,1         #x0 = x0 + 1
            add     $a1,$a1, $0         #y0 = y0
            li      $a2, YELLOW               #color = yellow
            jal     DrawCircle
            jal     Pause
            bgtu    $a0,492,reboundLeft       #Neu x0 = 512 - 20 = 492 tuc
den thanh phai thi thu hien bat trai
            j       Input
        right1:     #Thuc hien di chuyen sang phai
            li      $a2,0x00000000    #color = black
            jal     DrawCircle
            addi    $a0,$a0,1         #x0 = x0 + 1
            add     $a1,$a1, $0         #y0 = y0
            li      $a2, YELLOW               #color = yellow
            jal     DrawCircle
            jal     Pause1
            bgtu    $a0,492,reboundLeft       #Neu x0 = 512 - 20 = 492 tuc
den thanh phai thi thu hien bat trai
            j       Input
        right2:     #Thuc hien di chuyen sang phai
```

```
            li      $a2,0x00000000    #color = black
            jal     DrawCircle
            addi    $a0,$a0,1         #x0 = x0 + 1
            add     $a1,$a1, $0       #y0 = y0
            li      $a2, YELLOW               #color = yellow
            jal     DrawCircle
            jal     Pause2
            bgtu    $a0,492,reboundLeft       #Neu x0 = 512 - 20 = 492 tuc
den thanh phai thi thu hien bat trai
            j       Input
      up:   #Thuc hien di chuyen len tren
            move    $t9,$t0
            bgtz    $s1,up2
            bltz    $s1,up1
            li      $a2,0x00000000    #color = black
            jal     DrawCircle
            addi    $a1,$a1,-1        #y0 = y0 - 1
            add     $a0,$a0,$0        #x0 = x0
            li      $a2, YELLOW               #color = yellow
            jal     DrawCircle
            jal     Pause
            bltu    $a1,20,reboundDown        #Neu y0 = 20 tuc den
thanh tren cung thi thu hien bat xuong
            j       Input
      up1:  #Thuc hien di chuyen len tren
            li      $a2,0x00000000    #color = black
            jal     DrawCircle
            addi    $a1,$a1,-1   #y0 = y0 - 1
            add     $a0,$a0,$0        #x0 = x0
            li      $a2, YELLOW               #color = yellow
            jal     DrawCircle
            jal     Pause1
            bltu     $a1,20,reboundDown       #Neu y0 = 20 tuc den
thanh tren cung thi thu hien bat xuong
            j       Input
      up2:  #Thuc hien di chuyen len tren
```

```
            li      $a2,0x00000000    #color = black
            jal     DrawCircle
            addi    $a1,$a1,-1    #y0 = y0 - 1
            add     $a0,$a0,$0          #x0 = x0
            li      $a2, YELLOW               #color = yellow
            jal     DrawCircle
            jal     Pause2
            bltu    $a1,20,reboundDown          #Neu y0 = 20 tuc den
thanh tren cung thi thu hien bat xuong
            j       Input
        down:       #Thuc hien di chuyen xuong duoi
            move $t9,$t0
            bgtz    $s1,down2
            bltz    $s1,down1
            li      $a2,0x00000000    #color = black
            jal     DrawCircle
            addi    $a1,$a1,1           #y0 = y0 + 1
            add     $a0,$a0,$0          #x0 = x0
            li      $a2, YELLOW               #color = yellow
            jal     DrawCircle
            jal     Pause
            bgtu    $a1,492,reboundUp           #Neu y0 = 512 - 20 = 492
tuc den thanh duoi cung thi thu hien bat len
            j       Input
        down1:      #Thuc hien di chuyen xuong duoi
            li      $a2,0x00000000    #color = black
            jal     DrawCircle
            addi    $a1,$a1,1           #y0 = y0 + 1
            add     $a0,$a0,$0          #x0 = x0
            li      $a2, YELLOW               #color = yellow
            jal     DrawCircle
            jal     Pause1
            bgtu    $a1,492,reboundUp           #Neu y0 = 512 - 20 = 492
tuc den thanh duoi cung thi thu hien bat len
            j       Input
        down2:      #Thuc hien di chuyen xuong duoi
```

```
		li	$a2,0x00000000	#color = black
		jal	DrawCircle
		addi	$a1,$a1,1		#y0 = y0 + 1
		add	$a0,$a0,$0		#x0 = x0
		li	$a2, YELLOW		#color = yellow
		jal	DrawCircle
		jal	Pause2
		bgtu	$a1,492,reboundUp		#Neu y0 = 512 - 20 = 492 tuc den thanh duoi cung thi thu hien bat len
		j	Input
	reboundLeft:	#Thuc hien bat sang trai
		li	$t3 97	#Gan $t3 voi 'a' roi luu vao dia chi $k0
		sw	$t3,0($k0)
		j	Input
	reboundRight:	#Thuc hien bat sang phai
		li	$t3 100	#Gan $t3 voi 'd' roi luu vao dia chi $k0
		sw	$t3,0($k0)
		j	Input
	reboundDown:	#Thuc hien bat xuong duoi
		li	$t3 115	#Gan $t3 voi 's' roi luu vao dia chi $k0
		sw	$t3,0($k0)
		j	Input
	reboundUp: #Thuc hien bat len tren
		li	$t3 119	#Gan $t3 voi 'w' roi luu vao dia chi $k0
		sw	$t3,0($k0)
		j	Input
Input: #Thuc hien doc ki tu tu ban phim nhap vao bang cach luu vao thanh ghi $t0
	ReadKey:
		lw	$t0, 0($k0) # $t0 = [$k0] = KEY_CODE
	j moving
Pause:	#vi thanh ghi $a0 trung voi bien so x0 nen de syscall 32 thi phai su dung stack de luu tam thoi gia tri $a0
	addiu	$sp,$sp,-4
	sw	$a0, ($sp)
```

```
        li          $a0,5       # speed = 5ms
        li          $v0, 32         #syscall sleep
        syscall
        lw          $a0,($sp)       #tra lai gia tri $a0
        addiu       $sp,$sp,4
        jr          $ra
Pause1:     #vi thanh ghi $a0 trung voi bien so x0 nen de syscall 32 thi phai
su dung stack de luu tam thoi gia tri $a0
        addiu       $sp,$sp,-4
        sw          $a0, ($sp)
        li          $a0,10          # speed = 10ms
        li          $v0, 32         #syscall sleep
        syscall
        lw          $a0,($sp)       #tra lai gia tri $a0
        addiu       $sp,$sp,4
        jr          $ra
Pause2:     #vi thanh ghi $a0 trung voi bien so x0 nen de syscall 32 thi phai
su dung stack de luu tam thoi gia tri $a0
        addiu       $sp,$sp,-4
        sw          $a0, ($sp)
        li          $a0,0       # speed = 0ms
        li          $v0, 32         #syscall sleep
        syscall
        lw          $a0,($sp)       #tra lai gia tri $a0
        addiu       $sp,$sp,4
        jr          $ra
DrawCircle:#Using Midpoint Circle Algorithm
        #MAKE ROOM ON STACK
        addi    $sp, $sp, -20       #Make room on stack for 1 words
                sw      $ra, 0($sp)     #Store $ra on element 0 of stack
        sw      $a0, 4($sp)     #Store $a0 on element 1 of stack
        sw      $a1, 8($sp)     #Store $a1 on element 2 of stack
        sw      $a2, 12($sp)        #Store $a2 on element 3 of stack
        sw      $a3, 16($sp)        #Store $a3 on element 4 of stack
        #VARIABLES
        move    $t0, $a0        #x0
```

```
        move    $t1, $a1          #y0
        move    $t2, $a3          #radius
        addi    $t3, $t2, 0         #x
        li      $t4, 0          #y
        li      $t7, 0          #Err
        #While(x >= y)
circleLoop:
        blt       $t3, $t4, skipCircleLoop    #If x < y, skip circleLoop
                 #s5 = x, s6 = y
        #Draw Dot (x0 + x, y0 + y)
        addu      $s5, $t0, $t3
        addu      $s6, $t1, $t4
        lw        $a2, 12($sp)
        jal       drawDot          #Jump to drawDot
      #Draw Dot (x0 + y, y0 + x)
      addu      $s5, $t0, $t4
      addu      $s6, $t1, $t3
      lw      $a2, 12($sp)
      jal     drawDot          #Jump to drawDot
      #Draw Dot (x0 - y, y0 + x)
      subu      $s5, $t0, $t4
      addu      $s6, $t1, $t3
      lw      $a2, 12($sp)
      jal     drawDot          #Jump to drawDot
      #Draw Dot (x0 - x, y0 + y)
      subu      $s5, $t0, $t3
      addu      $s6, $t1, $t4
      lw      $a2, 12($sp)
      jal     drawDot          #Jump to drawDot
      #Draw Dot (x0 - x, y0 - y)
      subu      $s5, $t0, $t3
      subu      $s6, $t1, $t4
      lw      $a2, 12($sp)
      jal     drawDot          #Jump to drawDot
      #Draw Dot (x0 - y, y0 - x)
      subu      $s5, $t0, $t4
```

```
        subu    $s6, $t1, $t3
        lw      $a2, 12($sp)
        jal     drawDot         #Jump to drawDot
        #Draw Dot (x0 + y, y0 - x)
        addu    $s5, $t0, $t4
        subu    $s6, $t1, $t3
        lw      $a2, 12($sp)
        jal     drawDot         #Jump to drawDot
        #Draw Dot (x0 + x, y0 - y)
        addu    $s5, $t0, $t3
        subu    $s6, $t1, $t4
        lw      $a2, 12($sp)
        jal     drawDot         #Jump to drawDot
          #If (err <= 0)
          bgtz    $t7, doElse
          addi    $t4, $t4, 1     #y++
          sll $t8, $t4, 1                 #Bitshift y left 1
          addi $t8, $t8, 1          #2y + 1
          addu $t7, $t7, $t8        #Add  e + (2y + 1)
          j     circleContinue    #Skip else stmt
          #Else If (err > 0)
          doElse:
          addi    $t3, $t3, -1    #x--
          sll $t8, $t3, 1                 #Bitshift x left 1
          addi $t8, $t8, 1          #2x + 1
          subu $t7, $t7, $t8        #Subtract e - (2x + 1)
          j circleContinue
circleContinue:
          #LOOP
          j     circleLoop
          #CONTINUE
          skipCircleLoop:
          #RESTORE $RA
          lw      $ra, 0($sp)     #Restore $ra from stack
          addiu   $sp, $sp, 20    #Readjust stack
          jr $ra
```

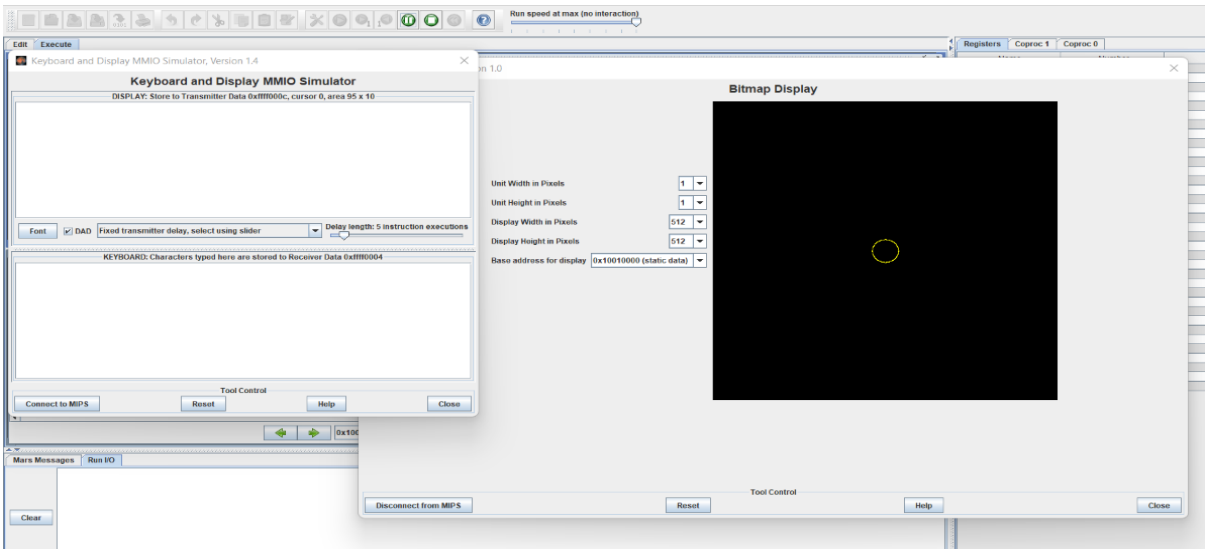
```
        nop
drawDot:
        add $at, $s6, $0
        sll    $at, $at, 9       # calculate offset in $at: at = y_pos * 512
        add    $at, $at, $s5      # at = y_pos * 512 + x_pos = "index"
        sll    $at, $at, 2       # at = (y_pos * 512 + x_pos)*4 = "offset"
        add    $at, $at, $v1      # at = v1 + offset
        sw     $a2, ($at)        # draw it!
        jr $ra
```

Tài liệu tham khảo :

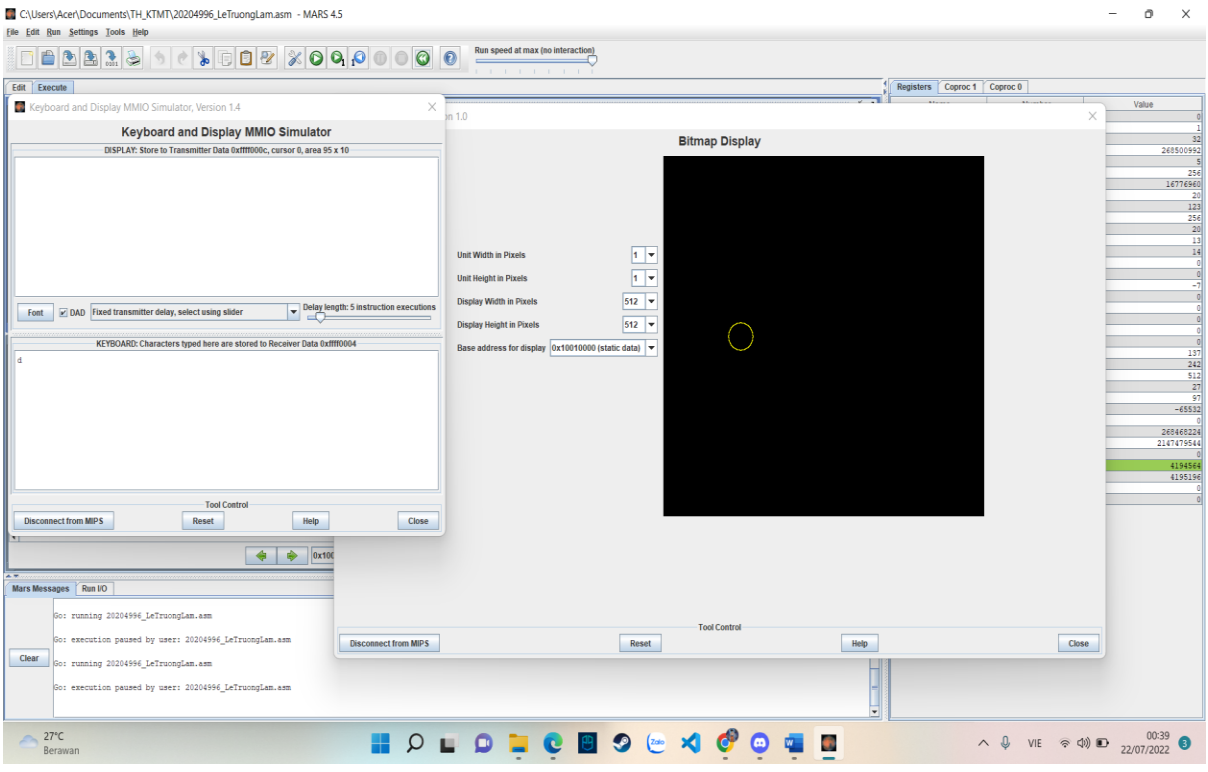
- Sử dụng thuật toán Midpoint Circle Algorithm để vẽ đường tròn
- Thuật toán vẽ đường tròn điểm giữa là một thuật toán được sử dụng để xác định các điểm cần thiết để tạo đường tròn. Sử dụng thuật toán điểm giữa để tính toán tất cả các điểm chu vi của hình tròn trong một bát phân đầu tiên và sau đó in chúng cùng với các điểm phản chiếu của chúng trong các bát phân khác. Điều này sẽ hiệu quả bởi vì một vòng tròn là đối xứng về tâm của nó.
- Code thuật toán

```
x = r;
y = 0;
e = 0;
while (x >= y){
    putpixel(x0 + x, y0 + y);
    putpixel(x0 + y, y0 + x);
    putpixel(x0 - y, y0 + x);
    putpixel(x0 - x, y0 + y);
    putpixel(x0 - x, y0 - y);
    putpixel(x0 - y, y0 - x);
    putpixel(x0 + y, y0 - x);
    putpixel(x0 + x, y0 - y);
    if (e <= 0){
        y = y + 1;
        e = e + 2y + 1;
    }
    else{
        x = x - 1;
        e = e - 2x - 1;
    }
}
```

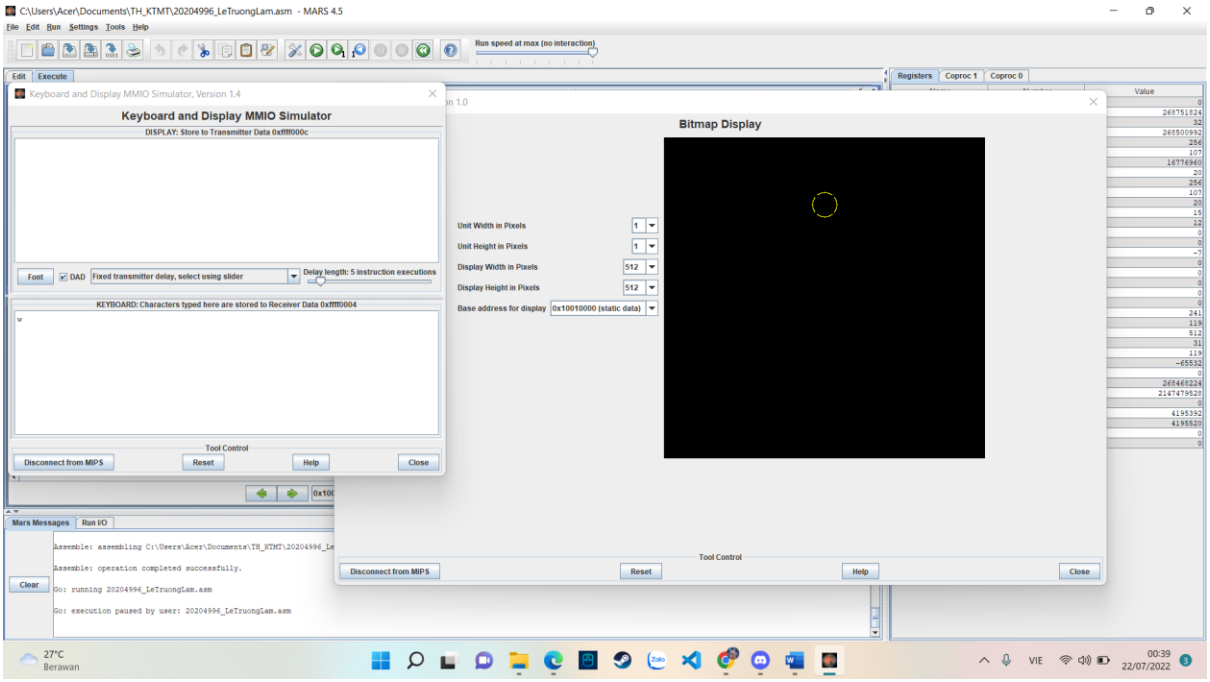
- Sử dụng các câu lệnh beq để thực hiện từng chức năng ‘a’,’s’,’w’,’d’,’z’,’x’. Ta thực hiện phép cộng trừ các tạo độ để tìm tâm mới của quả bóng và tô màu cho quả bóng tại vị trí mới. Quả bóng ở vị trí cũ sẽ tô màu trùng với màu nền.
- Tâm quả bóng cách các biên 1 khoảng bằng bán kính ta sẽ cho quả bóng di chuyển ngược lại



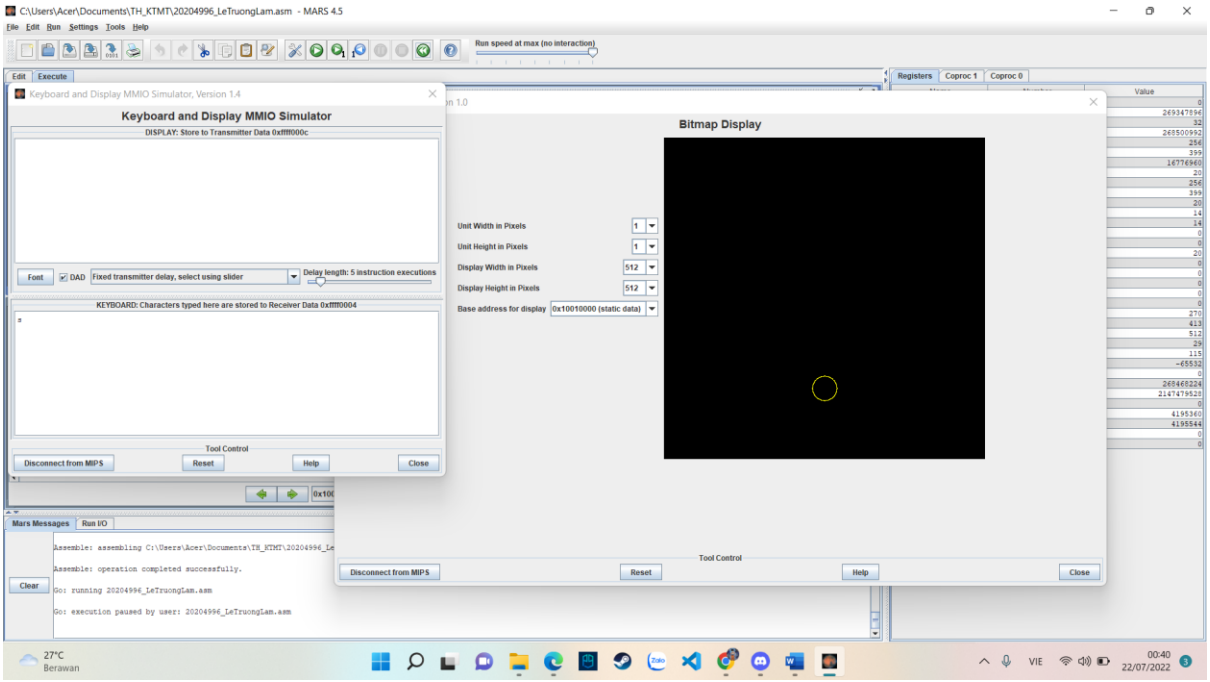
Khoi tao qua bong tai tam man hinh



Di chuyen sang trai



Di chuyen len tren



Di chuyen xuong duoi