

EXPERIMENT NO: ~~09~~ 08

(42)

Roll No: 5139

DATE: 13/5/22

Aim: To exchange data of two blocks using string instructions

THEORY:

PROGRAM

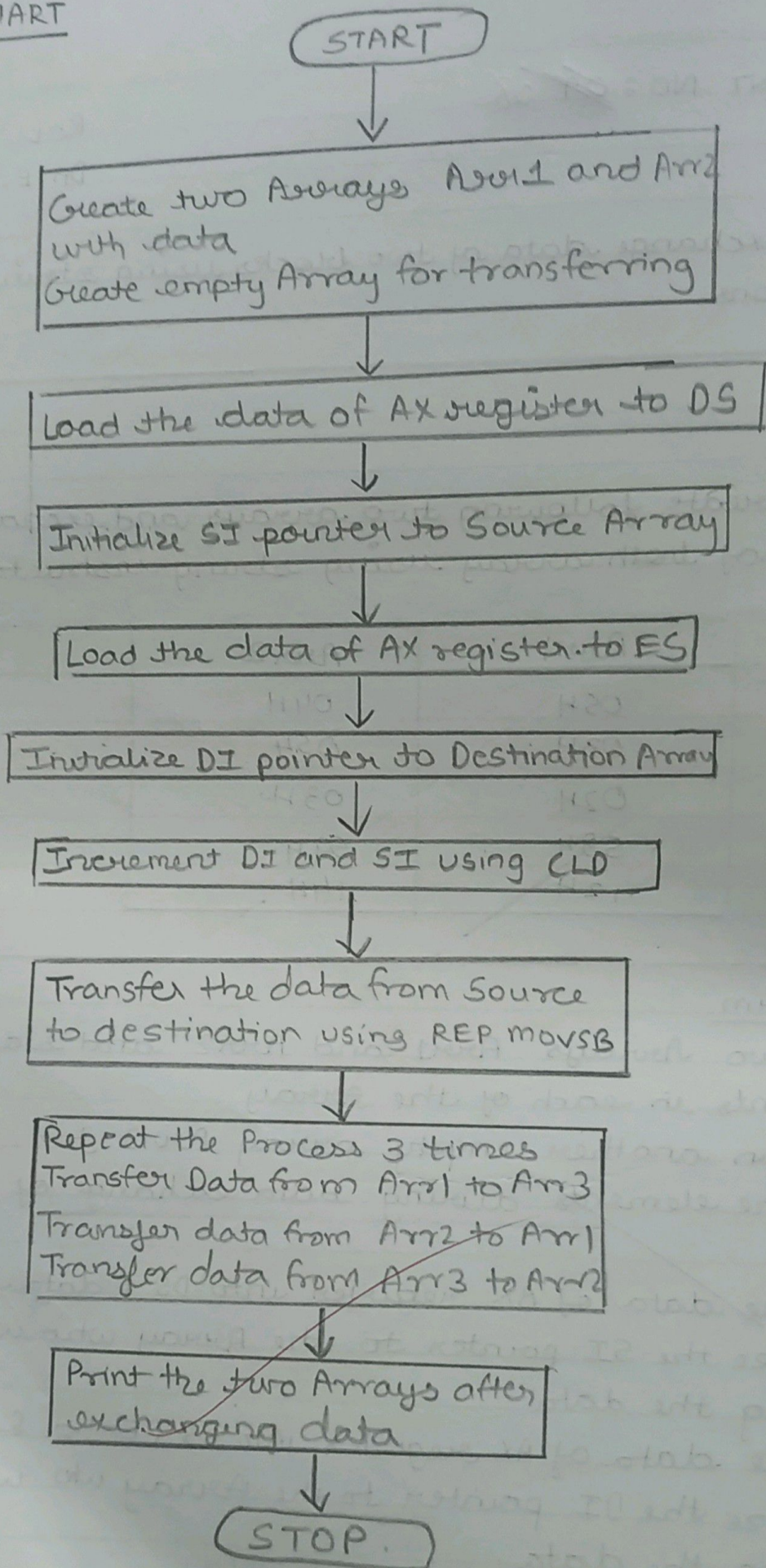
WAP to create following two arrays and exchange the elements of both array using string instructions

| Arr1 | Arr2 |
|------|------|
| 05H  | 04H  |
| 03H  | 05H  |
| 02H  | 03H  |
| 58H  | 59H  |
| 12H  | 14H  |

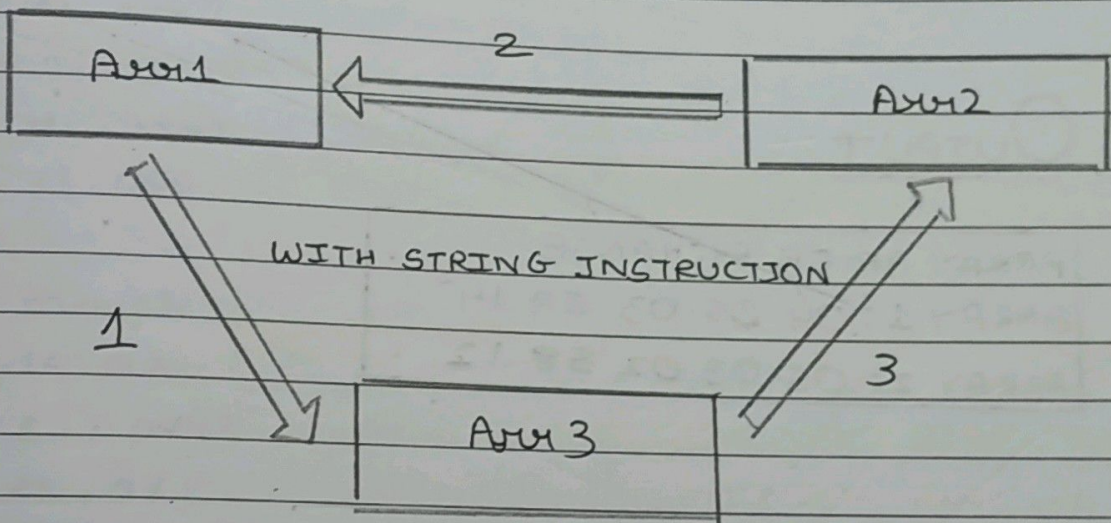
ALGORITHM

- 1] Create two Arrays Arr1 and Arr2 and store 5 elements in each of the Array.
- 2] Create an another empty array Arr3 to store the elements during data exchange of two blocks
- 3] Load the data of AX register into DS (data segment)
- 4] Initialize the SI pointer to the Array who will be sending the data
- 5] Load the data of AX register into ES (Extra Segment)
- 6] Initialize the DI pointer to the Array who will be receiving the data

# FLOWCHART



- 7] Increment SI and DI using CLO instruction
- 8] Transfer the data from source to destination using REP MOVSB instruction.
- 9] In this program, we need to repeat the process three times.
  - (i) Data from Arr1 will be transferred to Arr3
  - (ii) Data from Arr2 will be transferred to Arr1
  - (ii) Data from Arr3 will be transferred to Arr2



### PROGRAM

```

.model small
.data
arr1 db 05h, 03h, 02h, 58h, 12h
arr2 db 04h, 05h, 03h, 59h, 14h
arr3 db 5 dup(?)
str1 db 'ARRAY AFTER EXCHANGE', 0Ah, 0Dh, '$'
str2 db 'ARRAY 1:', '$'
str3 db 0Ah, 0Dh, 'ARRAY2:', '$'
.code

```

## STEPS TO DISPLAY OUTPUT

C:/> tasm filename.asm

C:/> link filename.obj

C:/> filename

## OUTPUT

ARRAY AFTER EXCHANGE

ARRAY 1: 04 05 03 59 14

ARRAY 2: 05 03 02 58 12

```
mov ax, @data
```

```
mov ds, ax
```

```
lea si, arr1
```

```
mov es, ax
```

```
lea di, arr3
```

```
mov cl, 05h
```

```
cld
```

```
rep movsb
```

```
mov ds, ax
```

```
lea si, arr2
```

```
mov es, ax
```

```
lea di, arr1
```

```
mov cl, 05h
```

```
cld
```

```
rep movsb
```

```
mov ds, ax
```

```
lea si, arr3
```

```
mov es, ax
```

```
lea di, arr2
```

```
cld
```

```
rep movsb
```

```
lea dx, str1
```

```
mov ah, 09h
```

```
int 21h
```

```
mov bl, 02h
```

```
pback: cmp bl, 01h
```

```
      jz pfront
```

```
      lea si, arr1
```

```
      lea dx, str2
```

```
      mov ah, 09h
```

```

int 21h
jmp pfront2
pfront: lea si, arr2
        lea dx, str3
        mov ah, 09h
        int 21h
pfront2: mov dh, 05h
back1:  mov al, [si]
        mov ch, 02H
        mov cl, 04H
        mov bh, al
I2:    rol bh, cl
        mov dl, bh
        and dl, 0Fh
        cmp dl, 09
        jbe I4
        add dl, 07H
I4:    add dl, 30H
        mov ah, 02H
        int 21H
        dec ch
        jnz I2
        mov dl, ''
        mov ah, 02h
        int 21H
        inc si
        dec dh
        jnz back1
        dec bl
        jnz pback

```

```
mov ah, 4ch  
int 21h  
END
```

CONCLUSION

Hence, we implemented the program to exchange data between two blocks using string instruction

~~Pratham~~  
28/5