



MICROPROCESSOR LABORATORY

Assignment No. 4



NAME :- OJUS PRAVIN JAISWAL

ROLL NO. :- SACO19108

DIVISION :- A

Assignment No. 4

Program :

```
%macro scall 4
```

```
    mov rax,%1
```

```
    mov rdi,%2
```

```
    mov rsi,%3
```

```
    mov rdx,%4
```

```
    syscall
```

```
%endmacro
```

```
%macro exit 0
```

```
    mov rax, 60
```

```
    mov rdi,0
```

```
    syscall
```

```
%endmacro
```

```
section .data
```

```
    arr dq
```

```
000000000000000100h,00000000000000004h,0000000000000003h,0000000000000002h,0000000000  
00001h
```

```
    n equ 5
```

```
menu db 10d,13d,"*****MENU*****"
```

```
    db 10d,13d,"1. Addition"
```

```
    db 10d,13d,"2. Subtraction"
```

```
    db 10d,13d,"3. Multiplication"
```

```
db 10d,13d,"4. Division"
```

```
db 10d,13d,"5. Exit"
```

```
db 10d,13d,"Enter your Choice : "
```

```
menu_len equ $-menu
```

```
m1 db 10d,13d,"Addition : "
```

```
l1 equ $-m1
```

```
m2 db 10d,13d,"Subtraction : "
```

```
l2 equ $-m2
```

```
m3 db 10d,13d,"Multiplication : "
```

```
l3 equ $-m3
```

```
m4 db 10d,13d,"Division : "
```

```
l4 equ $-m4
```

```
section .bss
```

```
answer resb 16
```

```
choice resb 2
```

```
section .text
```

```
global _start:
```

```
_start:
```

```
up:scall 1,1,menu,menu_len
```

```
scall 0,0,choice,2
```

```
cmp byte[choice],'1'
```

```
je case1
```

```
cmp byte[choice],'2'
```

```
je case2
cmp byte[choice], '3'
je case3
cmp byte[choice], '4'
je case4
cmp byte[choice], '5'
je case5
```

```
case1: scall 1,1,m1,l1
      call addition
      jmp up
```

```
case2: scall 1,1,m2,l2
      call subtraction
      jmp up
```

```
case3: scall 1,1,m3,l3
      call multiplication
      jmp up
```

```
case4: scall 1,1,m4,l4
      call division
      jmp up
```

```
case5: exit
```

```
addition:
      mov rcx, n
```

```
    dec rcx
    mov rsi,arr
    mov rax,[rsi]
up1:add rsi,8
    mov rbx,[rsi]
    add rax,rbx
    loop up1
    call display
ret
```

```
subtraction:
    mov rcx,n
    dec rcx
    mov rsi,arr
    mov rax,[rsi]
up2:add rsi,8
    mov rbx,[rsi]
    sub rax,rbx
    loop up2
    call display
ret
```

```
multiplication:
    mov rcx,n
    dec rcx
    mov rsi,arr
    mov rax,[rsi]
up3:add rsi,8
```

```
    mov rbx,[rsi]
    mul rbx
    loop up3
    call display
ret
```

division:

```
    mov rcx,n
    dec rcx
    mov rsi,arr
    mov rax,[rsi]
up4:add rsi,8
    mov rbx,[rsi]
    mov rdx,0
    div rbx
    loop up4
    call display
ret
```

display:

```
    mov rsi,answer+15
    mov rcx,16
cnt:mov rdx,0
    mov rbx,16
    div rbx
    cmp dl,09h
    jbe add30
    add dl,07h
```

add30:add dl,30h

mov [rsi],dl

dec rsi

dec rcx

jnz cnt

scall 1,1,answer,16

ret

```
</> Code  Input  Output  Run  Save
1  %macro scall 4
2      mov rax,%1
3      mov rdi,%2
4      mov rsi,%3
5      mov rdx,%4
6      syscall
7  %endmacro
8
9  %macro exit 0
10     mov rax, 60
11     mov rdi,0
12     syscall
13 %endmacro
14
15 section .data
16     arr dq 0000000000000100h,0000000000000004h,0000000000000003h,0000000000000002h,0000000000000001h
17     n equ 5
18
19     menu db 10d,13d,"*****MENU*****"
20         db 10d,13d,"1. Addition"
```

```
</> Code  Input  Output  Run  Save
21         db 10d,13d,"2. Subtraction"
22         db 10d,13d,"3. Multiplication"
23         db 10d,13d,"4. Division"
24         db 10d,13d,"5. Exit"
25         db 10d,13d,"Enter your Choice : "
26     menu_len equ $-menu
27
28     m1 db 10d,13d,"Addition : "
29     l1 equ $-m1
30     m2 db 10d,13d,"Subtraction : "
31     l2 equ $-m2
32     m3 db 10d,13d,"Multiplication : "
33     l3 equ $-m3
34     m4 db 10d,13d,"Division : "
35     l4 equ $-m4
36
37 section .bss
38     answer resb 16
39     choice resb 2
40
```

</> Code ≡ Input >_ Output

▶ Run

📄 Save

```
41 section .text
42 global _start:
43 _start:
44
45     up:scall 1,1,menu,menu_len
46         scall 0,0,choice,2
47
48     cmp byte[choice],'1'
49     je case1
50     cmp byte[choice],'2'
51     je case2
52     cmp byte[choice],'3'
53     je case3
54     cmp byte[choice],'4'
55     je case4
56     cmp byte[choice],'5'
57     je case5
58
59     case1: scall 1,1,m1,l1
60             call addition
```

</> Code ≡ Input >_ Output

▶ Run

📄 Save

```
61             jmp up
62
63     case2: scall 1,1,m2,l2
64             call subtraction
65             jmp up
66
67     case3: scall 1,1,m3,l3
68             call multiplication
69             jmp up
70
71     case4: scall 1,1,m4,l4
72             call division
73             jmp up
74
75     case5:exit
76
77 addition:
78     mov rcx,n
79     dec rcx
80     mov rsi,arr
```

</> Code ≡ Input >_ Output

▶ Run

📄 Save

```
81     mov rax,[rsi]
82 up1:add rsi,8
83     mov rbx,[rsi]
84     add rax,rbx
85     loop up1
86     call display
87     ret
88
89 subtraction:
90     mov rcx,n
91     dec rcx
92     mov rsi,arr
93     mov rax,[rsi]
94 up2:add rsi,8
95     mov rbx,[rsi]
96     sub rax,rbx
97     loop up2
98     call display
99     ret
100
```


</> Code ≡ Input >_ Output

▶ Run

📄 Save

```
101 multiplication:
102     mov rcx,n
103     dec rcx
104     mov rsi,arr
105     mov rax,[rsi]
106 up3: add rsi,8
107     mov rbx,[rsi]
108     mul rbx
109     loop up3
110     call display
111     ret
112
113 division:
114     mov rcx,n
115     dec rcx
116     mov rsi,arr
117     mov rax,[rsi]
118 up4: add rsi,8
119     mov rbx,[rsi]
120     mov rdx,0
```

</> Code ≡ Input >_ Output

▶ Run

📄 Save

```
121     div rbx
122     loop up4
123     call display
124     ret
125
126 display:
127     mov rsi,answer+15
128     mov rcx,16
129 cnt: mov rdx,0
130     mov rbx,16
131     div rbx
132     cmp dl,09h
133     jbe add30
134     add dl,07h
135 add30: add dl,30h
136     mov [rsi],dl
137     dec rsi
138     dec rcx
139     jnz cnt
140     scall 1,1,answer,16
```

</> Code ≡ Input >_ Output

▶ Run

📄 Save

```
122     loop up4
123     call display
124     ret
125
126 display:
127     mov rsi,answer+15
128     mov rcx,16
129 cnt: mov rdx,0
130     mov rbx,16
131     div rbx
132     cmp dl,09h
133     jbe add30
134     add dl,07h
135 add30: add dl,30h
136     mov [rsi],dl
137     dec rsi
138     dec rcx
139     jnz cnt
140     scall 1,1,answer,16
141     ret
```

Input :

</> Code

☰ Input

>_ Output

▶ Run

📄 Save

```
1 1
2 2
3 3
4 4
5 5
```

Output :

</> Code

☰ Input

>_ Output

▶ Run

📄 Save

```
*****MENU*****
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your Choice :
Addition : 00000000000010A
*****MENU*****
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your Choice :
Subtraction : 000000000000F6
*****MENU*****
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your Choice :
Multiplication : 000000000001800
```

Code Input Output

Run

Save

```
5. Exit
Enter your Choice :
Multiplication : 000000000001800
*****MENU*****
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your Choice :
Division : 00000000000000A
*****MENU*****
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your Choice :
[Program exited with exit code 0]
```