Practical4

```
section .data
   menu db 10, "-----MENU-----", 10
        db "1. Addition", 10
        db "2. Subtraction", 10
        db "3. Multiplication", 10
        db "4. Division", 10
        db "5. Exit", 10
        db "Enter your choice: ", 0
   menu_len equ $ - menu

   prompt1 db "Enter first number: ", 0
   prompt1_len equ $ - prompt1

   prompt2 db "Enter second number: ", 0
   prompt2_len equ $ - prompt2

   wch db "Invalid choice.", 10
   wch_len equ $ - wch

   addition db "Addition is: ", 0
   addition_len equ $ - addition

   subtract db "Subtraction is: ", 0
   subtract_len equ $ - subtract

   multiplication db "Multiplication is: ", 0
   multiplication_len equ $ - multiplication

   division db "Division is: ", 0
   division_len equ $ - division

   error db "Can't divide by zero", 10
   error_len equ $ - error

section .bss
   choice resb 2
   num1 resb 10
   num2 resb 10
   result resb 16

%macro disp 2
   mov rax, 1
   mov rdi, 1
   mov rsi, %1
   mov rdx, %2
   syscall
%endmacro
```

```
%macro read 2
   mov rax, 0
   mov rdi, 0
   mov rsi, %1
   mov rdx, %2
   syscall
%endmacro

section .text
   global _start

_start:
menu_loop:
   disp menu, menu_len
   read choice, 2
   mov bl, byte [choice]

   cmp bl, '1'
   je do_add
   cmp bl, '2'
   je do_sub
   cmp bl, '3'
   je do_mul
   cmp bl, '4'
   je do_div
   cmp bl, '5'
   je exit

   disp wch, wch_len
   jmp menu_loop

get_input:
   disp prompt1, prompt1_len
   read num1, 10
   call atoi
   mov r8, rax       ; store num1 in r8

   disp prompt2, prompt2_len
   read num2, 10
   call atoi
   mov r9, rax       ; store num2 in r9
   ret

do_add:
   call get_input
   mov rax, r8
   add rax, r9
   call dispproc_64
```

```
    disp addition, addition_len
    disp result, 16
    jmp menu_loop

do_sub:
    call get_input
    mov rax, r8
    sub rax, r9
    call dispproc_64
    disp subtract, subtract_len
    disp result, 16
    jmp menu_loop

do_mul:
    call get_input
    mov rax, r8
    mov rbx, r9
    mul rbx
    call dispproc_64
    disp multiplication, multiplication_len
    disp result, 16
    jmp menu_loop

do_div:
    call get_input
    mov rax, r8
    mov rbx, r9
    cmp rbx, 0
    je div_error
    xor rdx, rdx
    div rbx
    call dispproc_64
    disp division, division_len
    disp result, 16
    jmp menu_loop

div_error:
    disp error, error_len
    jmp menu_loop

exit:
    mov rax, 60
    xor rdi, rdi
    syscall

; ------------------------------------------
; Convert ASCII string to integer
; input: num1 or num2 in RSI
; output: RAX = integer value
```

```
; ------------------------------------------
atoi:
   xor rax, rax
   xor rcx, rcx
.next_digit:
   mov bl, byte [rsi + rcx]
   cmp bl, 10
   je .done
   cmp bl, 13
   je .done
   cmp bl, 0
   je .done
   cmp bl, '0'
   jl .done
   cmp bl, '9'
   jg .done
   sub bl, '0'
   imul rax, rax, 10
   add rax, rbx
   inc rcx
   jmp .next_digit
.done:
   ret

; ------------------------------------------
; Convert number to ASCII hex string
; Output in `result`
; ------------------------------------------
dispproc_64:
   mov rbx, rax
   mov rdi, result
   mov cx, 16
.next:
   rol rbx, 4
   mov al, bl
   and al, 0Fh
   cmp al, 9
   jbe .digit
   add al, 7
.digit:
   add al, 30h
   mov [rdi], al
   inc rdi
   dec cx
   jnz .next
ret
```

output

```
rllab@fedora:/home/liveuser$ nasm -f elf64 prathamesh4.nasm
rllab@fedora:/home/liveuser$ ld -o prathamesh4 prathamesh4.o
rllab@fedora:/home/liveuser$ ./prathamesh4

-----MENU-----
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 1
Enter first number: 64
Enter second number: 78
Addition is: 000000000000008E
-----MENU-----
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 2
Enter first number: 97
Enter second number: 41
Subtraction is: 0000000000000038
-----MENU-----
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 3
Enter first number: 97
Enter second number: 12
Multiplication is: 000000000000048C
-----MENU-----
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 4
Enter first number: 954
Enter second number: 2
Division is: 000000000000006F
-----MENU-----
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 5
rllab@fedora:/home/liveuser$
```