21CS8056
Sayan Paul

Compiler Design Lab-2

1. Write a C program that performs the following tasks using inline assembly:
a) Print a message to the console using the write system call.
b) Read an integer input from the user using the read system call.
c) Calculate the square of the input integer using inline assembly.
d) Print the result using the write system call.
Your program should interact with the user by prompting for input and displaying the results. Make sure to use appropriate registers for passing parameters and receiving results.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <ctype.h>

char ipMsg[] = "Enter n: ";
char sqMsg[] = "Square = %ld\n";

int main()
{
   long long n = 0;
   char s[17];
   write(1, ipMsg, strlen(ipMsg));
   read(0, s, 17);

   n = atoll(s);
```

```
    // square
    asm volatile(
        "movq %[num], %%rax\n\t"
        "imulq %%rax, %%rax\n\t"
        "movq %%rax, %[num]"
        : [num] "+rm"(n));

    char ans[256];
    sprintf(ans, sqMsg, n);
    write(1, ans, strlen(ans));

    return 0;
}
```

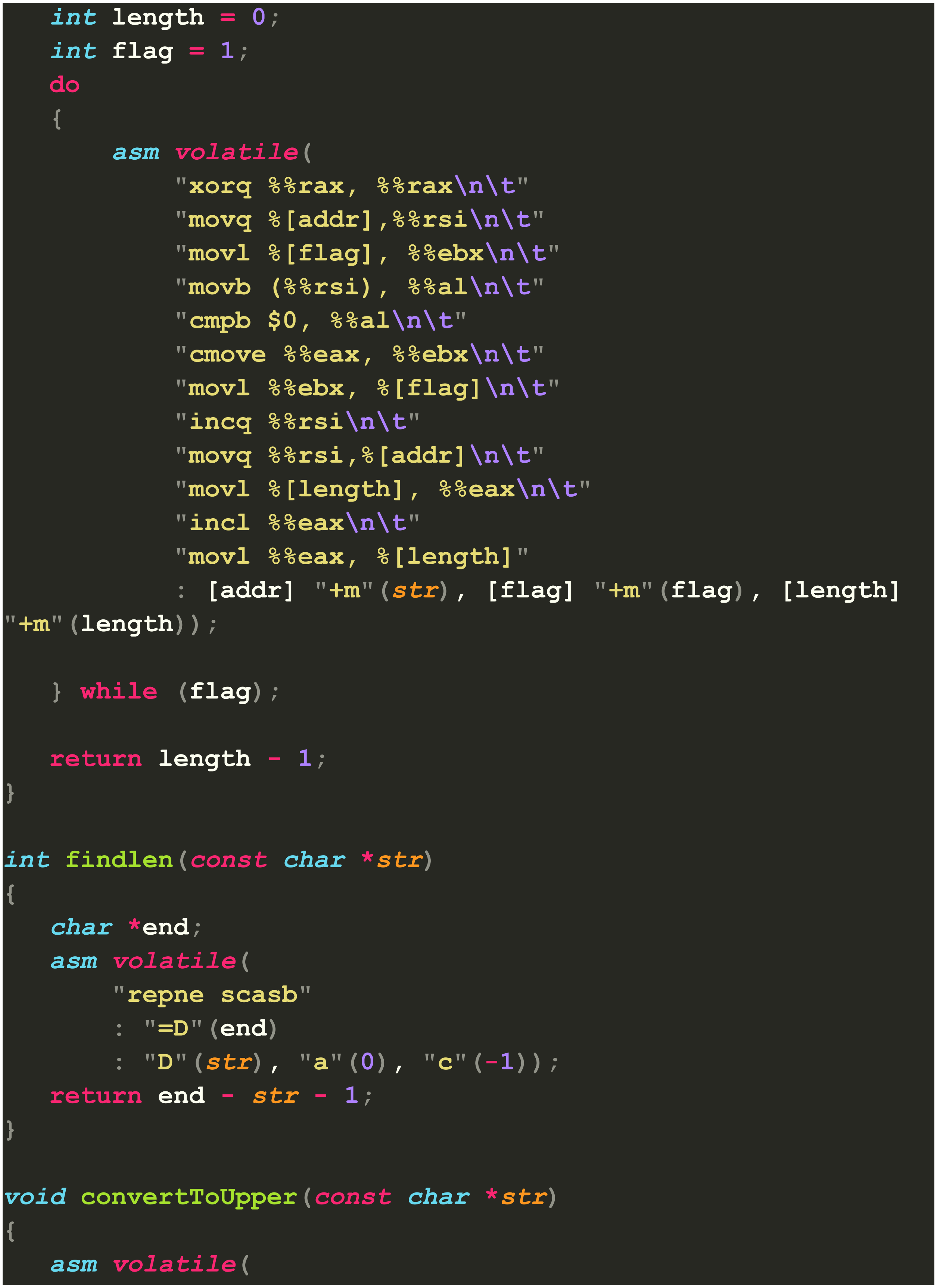
Output:

```
Enter n: 5
Square = 25
```

2. Write a C program that performs string manipulation using inline assembly:
a) Declare a string in C.
b) Calculate and print the length of the string using inline assembly.
c) Convert all lowercase characters in the string to uppercase using inline assembly.
d) Print the modified string.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int len(const char *str)
{
```



```
    int length = 0;
    int flag = 1;
    do
    {
        asm volatile(
            "xorq %%rax, %%rax\n\t"
            "movq %[addr],%%rsi\n\t"
            "movl %[flag], %%ebx\n\t"
            "movb (%%rsi), %%al\n\t"
            "cmpb $0, %%al\n\t"
            "cmove %%eax, %%ebx\n\t"
            "movl %%ebx, %[flag]\n\t"
            "incq %%rsi\n\t"
            "movq %%rsi,%[addr]\n\t"
            "movl %[length], %%eax\n\t"
            "incl %%eax\n\t"
            "movl %%eax, %[length]"
            : [addr] "+m"(str), [flag] "+m"(flag), [length]
"+m"(length));

    } while (flag);

    return length - 1;
}

int findlen(const char *str)
{
    char *end;
    asm volatile(
        "repne scasb"
        : "=D"(end)
        : "D"(str), "a"(0), "c"(-1));
    return end - str - 1;
}

void convertToUpper(const char *str)
{
    asm volatile(
        "cld\n"
        "1: lodsb\n"
        "testb %%al, %%al\n"
        "jz end\n"
        "cmpb $'a', %%al\n"
        "jb not_lower\n"
        "subb $32, %%al\n"
        "stosb\n"
        "jmp 1b\n"
        "not_lower: stosb\n"
        "jmp 1b\n"
        "end: nop\n"
        :
        : "S"(str), "D"(str)
        : "rax");
}

int main()
{
    char str[256];
    scanf("%s", str);
    printf("%ld\n", strlen(str));
    printf("%d\n", len(str));
    printf("%d\n", findlen(str));
    convertToUpper(str);
    printf("%s", str);

    return 0;
}
```

Output:

```
sayan
5
5
```

5

SAYAN