Procedure

In this assignment you need to do the following task:

a. First, ask the user for a choice between number 1 and 3. If the user provides any number other than 1, 2 and 3, print invalid choice and terminate your program.
b. If the choice is 1, print the star pattern as follows for input say, 4 (which could be anything less than 10)

```
*
**
***
****
…….
```

c. If the choice is 2, print the digit pattern as follows for input say, 5 (which could be anything less than 10)

```
1
12
123
1234
12345
…….
```

d. If the choice is 3, print the star pattern as follows for input say, 5 (which could be anything less than 10)

```
*****
****
***
**
*
```

However you need to use the procedure to perform your assignment.

<u>You need three procedures as star, digit and star-star combination</u>

So, what is a procedure?

Large problems can be divided into smaller tasks to make them more manageable
• A procedure is the ASM equivalent of a Java or C++ function
• Following is an assembly language procedure named Sample:

```
Sample:
   …..
   …..
   ret
```

A description of all tasks accomplished by the procedure.
•Receives: A list of input parameters; state their usage and requirements.
• Returns: A description of values returned by the procedure.
• Requires: Optional list of requirements called preconditions that must be satisfied before the procedure is called.

The CALL instruction calls a procedure
• pushes offset of next instruction on the stack
• copies the address of the called procedure into EIP
• The RET instruction returns from a procedure
• pops top of stack into EIP

A sample example

Example
Let us write a very simple procedure named sum that adds the variables stored in the ECX and EDX register and returns the sum in the EAX register −

```
section    .text
   global _start        ;must be declared for using gcc

_start:                    ;tell linker entry point
   mov     ecx,'4'
   sub     ecx, '0'

   mov     edx, '5'
   sub     edx, '0'

   call    sum          ;call sum procedure
   mov     [res], eax
   mov     ecx, msg
   mov     edx, len
   mov     ebx,1        ;file descriptor (stdout)
   mov     eax,4        ;system call number (sys_write)
   int     0x80         ;call kernel

   mov     ecx, res
   mov     edx, 1
   mov     ebx, 1             ;file descriptor (stdout)
```

```
   mov     eax, 4              ;system call number (sys_write)
   int     0x80           ;call kernel

   mov     eax,1          ;system call number (sys_exit)
   int     0x80           ;call kernel
sum:
   mov      eax, ecx
   add      eax, edx
   add      eax, '0'
   ret

section .data
msg db "The sum is:", 0xA,0xD
len equ $- msg

segment .bss
res resb 1
```

Command line will ask for the choice from the user. Please see the following input/output scenario for clarification

Input	Output
Please enter your choice: 1. Star 2. Digit 3. Star Star Combination 1 Enter number 3	* ** ***
Please enter your choice: 1. Star 2. Digit 3. Star Star Combination 2 Enter number 7	1 12 123 1234 12345 123456 1234567
Please enter your choice: 1. Star 2. Digit 3. Star Star Combination	*** ** *

3 Enter number 3	