1.

a. Write and execute assembly program to illustrate push and pop instructions

```
 PRESERVE8
        THUMB
        AREA |.text|,CODE,READONLY
        EXPORT __main
__main
        MOV R0,#5
        PUSH {R0}
        POP {R1}
stop    B       stop
END
```

b. Write a C program to light a LED connected to port C15 using an interrupt.

```
    #include <stdio.h>
    #include "NUC1xx.h"
    #include "Driver\DrvGPIO.h"
    #include "Driver\DrvUART.h"
    #include "Driver\DrvSYS.h"

    void EINT1Callback(void)
    {  DrvGPIO_ClrBit(E_GPC, 15);
    DrvSYS_Delay(100000);
    DrvGPIO_SetBit(E_GPC, 15);
    DrvSYS_Delay(100000);
    }

    int main (void)
    { UNLOCKREG();
    DrvSYS_Open(48000000);
    LOCKREG();
    DrvGPIO_Open(E_GPC, 15, E_IO_OUTPUT);
    DrvGPIO_Open(E_GPB, 15, E_IO_INPUT);
    DrvGPIO_EnableEINT1(E_IO_BOTH_EDGE, E_MODE_EDGE,
    EINT1Callback);
    while(1) { }
}
```

2. **a.** Write and execute assembly program to clear the bits from 15 to 18 and extract the bits from 24 to 26 in a 32 bit data.

```
PRESERVE8
        THUMB
        AREA |.text|,CODE,READONLY
        EXPORT __main
__main
        LDR R0,=0xFFFFFFFF
        LDR R1,=0xFFFFFFFF
        MOVS R3,#15
        MOVS R4,#13
        RORS R0,R0,R3
        LSRS R0,R0,#4
        RORS R0,R0,R4
        LSLS R1,R1,#5
        LSRS R1,R1,#29
stop B stop
        END
```

b. Write a C program to scan the key board and display the key that is pressed on LCD or 7-segment.

```
#include <stdio.h>


#include "NUC1xx.h"
#include "DrvSYS.h"
#include "DrvGPIO.h"
#include "scankey.h"
#include "NUC1xx-LB_002\LCD_Driver.h"
```

```
int main (void)
{int8_t number;
char TEXT0[16]="Smpl_LCD_Keypad";
char TEXT1[16]="Keypad:        ";
UNLOCKREG();
DrvSYS_Open(48000000);
LOCKREG();
Initial_panel();
clr_all_panel();
OpenKeyPad();
print_lcd(0,TEXT0);
 while(1)
{number = Scankey();
sprintf(TEXT1+8,"%d",number);
print_lcd(1, TEXT1);
DrvSYS_Delay(5000);

}
}
```

3. **a.** Write and execute assembly program to swap 10 bytes of data, the locations starts from 20000100 to 20000200.

```
PRESERVE8
        THUMB
        AREA |.text|,CODE,READONLY
        EXPORT __main
__main
        LDR R0,=0x20000100
        LDR R1,=0x20000200
        MOV R2,#0
swap
        LDRB R3,[R0,R2]
        LDRB R4,[R1,R2]
        STRB R3,[R1,R2]
        STRB R4,[R0,R2]
        ADD R2,R2,#1
        CMP R2,#10
        BLT swap
stop B stop
        END
```

b. Write a C program to light a LED connected to port C14 using an interrupt.

```
#include <stdio.h>
#include "NUC1xx.h"
#include "Driver\DrvGPIO.h"
#include "Driver\DrvUART.h"
#include "Driver\DrvSYS.h"

void EINT1Callback(void)
{  DrvGPIO_ClrBit(E_GPC, 15);
DrvSYS_Delay(100000);
DrvGPIO_SetBit(E_GPC, 15);
DrvSYS_Delay(100000);
}

int main (void)
```

```
{ UNLOCKREG();
DrvSYS_Open(48000000);
LOCKREG();
DrvGPIO_Open(E_GPC, 15, E_IO_OUTPUT);
DrvGPIO_Open(E_GPB, 15, E_IO_INPUT);
DrvGPIO_EnableEINT1(E_IO_BOTH_EDGE, E_MODE_EDGE, EINT1Callback);
while(1) { }
}
```

4. **a.** Write and execute assembly program to illustrate the usage of functions (in UDF calculate y=2x+9).

```
PRESERVE8
        THUMB
        AREA |.text|,CODE,READONLY
        EXPORT __main
func
        MOVS R1,#2
        MOVS R3,#9
        MULS R0,R1,R0
        ADDS R0,R0,R3
        BX lr
__main
        MOV R0,#4
        BL func
stop B stop
        END
```

b. Write a C program to toggle RGB LEDs in sequence.

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

```
#include "NUC1xx.h"

#include "Driver\DrvGPIO.h"

#include "Driver\DrvUART.h"

#include "Driver\DrvSYS.h"


void Init_LED()

{DrvGPIO_Open(E_GPA, 12, E_IO_OUTPUT);

DrvGPIO_Open(E_GPA, 13, E_IO_OUTPUT);

DrvGPIO_Open(E_GPA, 14, E_IO_OUTPUT);

DrvGPIO_SetBit(E_GPA, 12);

DrvGPIO_SetBit(E_GPA, 13);

DrvGPIO_SetBit(E_GPA, 14);

}

int main (void)

{UNLOCKREG();

DrvSYS_Open(48000000);

LOCKREG();

Init_LED();

 while(1)

         {

   DrvGPIO_ClrBit(E_GPA,12);

         DrvSYS_Delay(1000000);

   DrvGPIO_SetBit(E_GPA,12);

   DrvGPIO_ClrBit(E_GPA,13);

         DrvSYS_Delay(1000000);

   DrvGPIO_SetBit(E_GPA,13);

   DrvGPIO_ClrBit(E_GPA,14);

         DrvSYS_Delay(1000000);
```

```
  DrvGPIO_SetBit(E_GPA,14);
        DrvSYS_Delay(1000000);


        }
}
```

5. **a.** Write and execute assembly program to find factorial of N number.

```
PRESERVE8
      THUMB
      AREA    |.text|, CODE, READONLY
         EXPORT       __main
__main
         MOVS R0,#5
      MOVS    R1,#1
loop
         MULS R1,R0,R1
         SUBS  R0,R0,#1
         BNE loop
         NOP
         END
```

b. Write a C program to display a string on LCD when interrupt ‘A’ arrives and display on 7-segment a digit ‘77’ when interrupt ‘B’ arrives.

```
#include <stdio.h>
#include "NUC1xx.h"
#include "Driver\DrvUART.h"
#include "Driver\DrvGPIO.h"
#include "Driver\DrvSYS.h"
#include "LCD_Driver.h"
```

```
#include "Seven_Segment.h"

void EINT0Callback()

{

            print_lcd(1,"Interrupt A");

}

void EINT1Callback()

{

                        print_lcd(3,"Interrupt B");

                        close_seven_segment();

                        show_seven_segment(0,7);

                        show_seven_segment(1,7);

                        DrvSYS_Delay(500000);

}

int main()

{

            char TEXT[16];

            UNLOCKREG();

            DrvSYS_Open(48000000);

            LOCKREG();

            Initial_panel();

            clr_all_panel();

            print_lcd(0,"Smpl_GPIO_Intr");

            DrvGPIO_Open(E_GPA, 14, E_IO_INPUT);

            DrvGPIO_EnableEINT0(E_IO_BOTH_EDGE, E_MODE_EDGE, EINT0Callback);

            DrvGPIO_Open(E_GPB, 15, E_IO_INPUT);

            DrvGPIO_EnableEINT1(E_IO_BOTH_EDGE, E_MODE_EDGE, EINT1Callback);
```

```
        while(1)
        {
        }
}
```

6.

a. Write and execute assembly program to find sum of N array elements, the array locations starts from 20000100 and store the result at the address 20000500.

```
PRESERVE8
        THUMB
        AREA |.text|,CODE,READONLY
        EXPORT __main
__main
        LDR R0,=0x20000100
        LDR R1,=0x20000500
        MOVS R2,#0
        MOVS R4,#0
sum
        LDRB R3,[R0,R2]
        ADDS R4,R4,R3
        ADDS R2,R2,#1
        CMP R2,#10
        BLT sum
        STR R4,[R1,#0]
stop B stop
        END
```

b. Write a C program to display Ring counter on LEDs.

```
#include <stdio.h>
#include "NUC1xx.h"
```

```
#include "Driver\DrvGPIO.h"
#include "Driver\DrvUART.h"
#include "Driver\DrvSYS.h"
int main (void)
{
    UNLOCKREG();
    DrvSYS_Open(48000000);
    LOCKREG();
    DrvGPIO_Open(E_GPC, 15, E_IO_OUTPUT);
    DrvGPIO_Open(E_GPC, 14, E_IO_OUTPUT);
    DrvGPIO_Open(E_GPC, 13, E_IO_OUTPUT);
    DrvGPIO_Open(E_GPC, 12, E_IO_OUTPUT);
    DrvGPIO_SetBit(E_GPC, 15);
    DrvGPIO_SetBit(E_GPC, 14);
    DrvGPIO_SetBit(E_GPC, 13);
    DrvGPIO_SetBit(E_GPC, 12);
    while(1)
    {
        DrvGPIO_ClrBit(E_GPC, 15);
        DrvSYS_Delay(100000);
        DrvGPIO_SetBit(E_GPC, 15);
        //DrvSYS_Delay(100000);
        DrvGPIO_ClrBit(E_GPC, 14);
        DrvSYS_Delay(100000);
        DrvGPIO_SetBit(E_GPC, 14);
        //DrvSYS_Delay(100000);
        DrvGPIO_ClrBit(E_GPC, 13);
        DrvSYS_Delay(100000);
```

```
                    DrvGPIO_SetBit(E_GPC, 13);

                    //DrvSYS_Delay(100000);

                    DrvGPIO_ClrBit(E_GPC, 12);

                    DrvSYS_Delay(100000);

                    DrvGPIO_SetBit(E_GPC, 12);

                    //DrvSYS_Delay(100000);

          }

}
```

7. **a.** Write and execute assembly program to illustrate nested functions.

```
PRESERVE8

          THUMB

          AREA |.text|,CODE,READONLY

          EXPORT func

          EXTERN func1

func

          MOVS R1,#5

          BL func1

stop B stop

          END

+++++++++++++++++++++++++

          PRESERVE8

          THUMB

          AREA |.text|,CODE,READONLY

          EXPORT func1

func1

          MOVS R0,#10
```

```
        BX LR
stop B stop
        END


+++++++++++++++++++++++++++++++
        PRESERVE8
        THUMB
        AREA |.text|,CODE,READONLY
        EXPORT __main
        EXTERN func
__main
        MOVS R0,#1
        BL func
stop B stop
        END
```

b. Write a C program to display a string “welcome to msrit” on LCD interface.

```
#include <stdio.h>
#include "NUC1xx.h"
#include "Driver\DrvSYS.h"
#include "Driver\DrvGPIO.h"
#include "NUC1xx-LB_002\LCD_Driver.h"


int main(void)
{
        UNLOCKREG();
        DrvSYS_Open(48000000);
        LOCKREG();


        Initial_panel();
        clr_all_panel();


        print_lcd(0, "Welcome to MSRIT   ");
}
```

8. **a.** Write and execute assembly program to implement switch-case.

```
PRESERVE8
        THUMB
        AREA |.text|,CODE,READONLY
        EXPORT __main
__main
        MOVS R0,#3
        CMP R0,#3
        BHI default_case
        MOVS R2,#4
        MUL R0,R2,R0
        LDR R1,=BranchTable
        LDR R2,[R1,R0]
        BX R2
        ALIGN 4
BranchTable
        DCD Test0
        DCD Test1
        DCD Test2
        DCD Test3
```

```
default_case
        MOVS R0,#1
stop B stop
Test0
        MOVS R0,#2
stop1 B stop1
Test1
        MOVS r0,#3
stop2 B stop2
Test2
        MOVS R0,#4
stop3 B stop3
Test3
        MOVS R0,#5
stop4 B stop4
        END
```

b. Write a C program to beep a buzzer connected to port b11 using an interrupt.

```
#include <stdio.h>
#include "NUC1xx.h"
#include "Driver\DrvGPIO.h"
#include "Driver\DrvUART.h"
#include "Driver\DrvSYS.h"


void EINT1Callback(void)
{ DrvGPIO_ClrBit(E_GPB,11);
DrvSYS_Delay(1000);
DrvGPIO_SetBit(E_GPB,11);
DrvSYS_Delay(1000);
}
int main(void)
{ UNLOCKREG();
DrvSYS_Open(48000000);
LOCKREG();
DrvGPIO_Open(E_GPB, 11, E_IO_OUTPUT);
DrvGPIO_Open(E_GPB, 15, E_IO_INPUT);
DrvGPIO_EnableEINT1(E_IO_BOTH_EDGE, E_MODE_EDGE, EINT1Callback);
while(1) { }
}
```

9. **a.** Write and execute assembly program to compute $y=2x^2+x+3$.

```
PRESERVE8
        THUMB
        AREA |.text|,CODE,READONLY
        EXPORT __main
__main
        MOVS R0,#5
        MOVS R1,#5
        MOVS R2,#2
        MUL R1,R1,R0
        MUL R1,R2,R1
        ADDS R1,R1,R0
        ADDS R1,R1,#3
stop B stop
        END
```

b. Write a C program to scan the keyboard and to beep buzzer (b11) when key 3 is pressed and light LED (c12) for a brief period when key 2 is pressed.

```
include <stdio.h>
#include "NUC1xx.h"
#include "DrvSYS.h"
#include "DrvGPIO.h"
#include "scankey.h"
#include "NUC1xx-LB_002\LCD_Driver.h"

int main (void)
{ int8_t number;
char TEXT0[16]="Smpl_LCD_Keypad";
char TEXT1[16]="Keypad:        ";
UNLOCKREG();
DrvSYS_Open(48000000);
LOCKREG();
Initial_panel();
clr_all_panel();
OpenKeyPad();
print_lcd(0,TEXT0);
DrvGPIO_Open(E_GPB, 11, E_IO_OUTPUT);
DrvGPIO_Open(E_GPC, 12, E_IO_OUTPUT);
DrvGPIO_SetBit(E_GPC, 12);
 while(1)
 {   number = Scankey();
 sprintf(TEXT1+8,"%d",number);
 print_lcd(1, TEXT1);
 DrvSYS_Delay(5000);
if(number==3)
{ DrvGPIO_ClrBit(E_GPB,11);
DrvSYS_Delay(1000);
DrvGPIO_SetBit(E_GPB,11);
DrvSYS_Delay(1000);
}
else if(number==2)
{ DrvGPIO_ClrBit(E_GPC, 12);
DrvSYS_Delay(3000000);
DrvGPIO_SetBit(E_GPC, 12);
DrvSYS_Delay(3000000);
}
}
}
```

10. **a.** Write and execute assembly program to copy 10 bytes, the source address starts from 20000000 and the destination address starts from 20000200.

```
PRESERVE8
        THUMB
        AREA |.text|,CODE,READONLY
        EXPORT __main
__main
        LDR R0,=0x20000000
        LDR R1,=0x20000200
        MOV R2,#0
swap
        LDRB R3,[R0,R2]
        STRB R3,[R1,R2]
        ADD R2,R2,#1

        CMP R2,#10

        BLT swap

stop B stop

        END
```

b. Write a C program to light a LED connected to port C12 using an interrupt.

```
#include <stdio.h>

#include "NUC1xx.h"

#include "Driver\DrvSYS.h"

#include "Seven_Segment.h"

#include "DrvGPIO.h"

#include "DrvSYS.h"

void seg_display(int16_t value)

{

        int8_t digit;

        digit = value / 1000;

        close_seven_segment();

        show_seven_segment(3,digit);

        DrvSYS_Delay(5000);

        value = value - digit * 1000;

        digit = value / 100;

        close_seven_segment();

        show_seven_segment(2,digit);

        DrvSYS_Delay(5000);

        value = value - digit * 100;

        digit = value / 10;

        close_seven_segment();
```

```
    show_seven_segment(1,digit);
    DrvSYS_Delay(5000);
    value = value - digit * 10;
    digit = value;
    close_seven_segment();
    show_seven_segment(0,digit);
    DrvSYS_Delay(5000);
}
int32_t main (void)
{
    int val = 0000;
    UNLOCKREG();
    DrvSYS_Open(48000000);
    LOCKREG();
    while(1)
    {
        DrvSYS_Delay(500);
        val++;
        seg_display(val);
        if(val==9999)
            val=0000;
    }
}
```