1. AREA MULTIPLY,CODE,READONLY
ENTRY
START
MOV R1,#20
MOV R2,#220
MUL R3,R1,R2
NOP
NOP
NOP
END

2. AREA SUM, CODE, READONLY
ENTRY
MOV R1,#10
MOV R2,#0
LOOP
ADD R2, R2, R1
SUBS R1,#0x01
BNE LOOP
BACK B BACK
END

3. AREA FACTORIAL, CODE, READONLY
ENTRY
START
MOV R0, #7
MOV R1, R0
FACT SUBS R1, R1, #1
CMP R1,#1;
BEQ STOP
MUL R3,R0,R1
MOV R0,R3
BNE FACT
STOP B ATOP
END

4. AREA ADDITION , CODE, READONLY
ENTRY
START
MOV R5,#6
MOV R0,#0
LDR R1,=VALUE1
LOOP
LDRH R3,[R1],#02
ADD R0,R0,R3
SUBS R5,R5,#1
CMP R5,#0
BNE LOOP
LDR R4,=RESULT
STR R0,[R4]
JMP B JMP

```
VALUE1 DCW 0X1111, 0X2222, 0X3333, 0XAAAA, 0XBBBB, 0XCCCC
AREA DATA2, DATA, READWRITE
RESULT DCD 0X0
        END
```

5.

```
        AREA SQUARE, CODE, READONLY ENTRY
START
        LDR R0, = TABLE1
        LDR R1,=8
        MOV R1, R1, LSL#0X2
        ADD R0, R0, R1
        LDR R3, [R0]
         NOP
        NOP
        NOP
TABLE1          DCD 0X00000000
                DCD 0X00000001
                DCD 0X00000004
                DCD 0X00000009
                DCD 0X00000010
                DCD 0X00000019
                DCD 0X00000024
                DCD 0X00000031
                DCD 0X00000040
                DCD 0X00000051
                DCD 0X00000064
                END
```

6.

```
AREA SMALLEST, CODE, READONLY

ENTRY

START
        MOV R5,#6
        LDR R1,=VALUE1
         LDR R2,[R1],#4
LOOP
         LDR R4,[R1],#4
        CMP R2,R4
        BLS LOOP1
         MOV R2,R4
LOOP1
        SUBS R5,R5,#1
        CMP R5,#0
         BNE LOOP
         LDR R4,=RESULT
         STR R2,[R4]
         NOP
         NOP
        NOP
VALUE1
```

```
        DCD 0X44444444
        DCD 0X22222222
        DCD 0X11111111
        DCD 0X33333333
        DCD 0XAAAAAAAA
        DCD 0X88888888
        DCD 0X99999999

AREA DATA2, DATA, READWRITE

RESULT DCD 0X0

        END
```

7.
```
#include <LPC213x.h>
#include <stdint.h>

void UART0_init(void)
{
   PINSEL0 = PINSEL0 | 0x00000005;
   U0LCR = 0x83;
   U0DLM = 0x00;
   U0DLL = 0x61;
   U0LCR = 0x03;
}

unsigned char UART0_RxChar(void)
{
   while((U0LSR & 0x01) == 0);
   return U0RBR;
}

void UART0_TxChar(char ch)
{
   U0THR = ch;
   while((U0LSR & 0x60) == 0);
}

void UART0_SendString(char *p)
{
   char c;
   while(*p != '\0')
   {
      c = *p;
      p++;
      UART0_TxChar(c);
   }
}

int main(void)
{
```

```
    char receive;
    UART0_init();
    while(1)
    {
        receive = UART0_RxChar();
        UART0_SendString("Received :");
        UART0_TxChar(receive);
        UART0_SendString("\r\n");
    }
}
```

8.
```
#include <LPC21xx.H>

void clock_wise(void);
void anti_clock_wise(void);

unsigned long int var1, var2;
unsigned int i = 0, j = 0, k = 0;

int main(void)
{
    PINSEL0 = 0x00FFFFFF;
    IO0DIR |= 0x0000F000;

    while(1)
    {
        for(j = 0; j < 50; j++)
            clock_wise();

        for(k = 0; k < 65000; k++);

        for(j = 0; j < 50; j++)
            anti_clock_wise();

        for(k = 0; k < 65000; k++);
    }
}

void clock_wise(void)
{
    var1 = 0x00000800;
    for(i = 0; i <= 3; i++)
    {
        var1 = var1 << 1;
        var2 = ~var1;
        var2 = var2 & 0x0000F000;
        IO0PIN = ~var2;
        for(k = 0; k < 3000; k++);
    }
}
```

```
void anti_clock_wise(void)
{
   var1 = 0x00010000;
   for(i = 0; i <= 3; i++)
   {
      var1 = var1 >> 1;
      var2 = ~var1;
      var2 = var2 & 0x0000F000;
      IO0PIN = ~var2;
      for(k = 0; k < 3000; k++);
   }
}
```

9.
```
#include <LPC21XX.h>

unsigned int delay;
unsigned int Switchcount = 0;
unsigned int Disp[16] = {
   0x003F0000, 0x00060000, 0x005B0000, 0x004F0000,
   0x00660000, 0x006D0000, 0x007D0000, 0x00070000,
   0x007F0000, 0x006F0000, 0x00770000, 0x007C0000,
   0x00390000, 0x005E0000, 0x00790000, 0x00710000
};

#define SELDISP1 0x10000000
#define SELDISP2 0x20000000
#define SELDISP3 0x40000000
#define SELDISP4 0x80000000
#define ALLDISP  0xF0000000
#define DATAPORT 0x00FF0000

int main(void)
{
   PINSEL0 = 0x00000000;
   PINSEL1 = 0x00000000;
   IO0DIR = 0xF0FF0000;
   IO1DIR = 0x00000000;

   while(1)
   {
      IO0SET |= ALLDISP;
      IO0CLR = 0x00FF0000;
      IO0SET = Disp[Switchcount];

      if(!(IO1PIN & 0x00800000))
      {
         for(delay = 0; delay < 100000; delay++);

         if((IO1PIN & 0x00800000))
```

```
        {
          Switchcount++;
          if(Switchcount == 0x10)
          {
            Switchcount = 0;
            IO0CLR = 0xF0FF0000;
          }
        }
      }
    }
}
```