ADDITION AND MULTIPLICATION AND SUBSTRACTION OF TWO 32-BIT NUMBERS

AREA ADDITION, CODE, READONLY

ENTRY

MOV R0,#0X000000FF

MOV R1,#0X000000CD

ADD R2, R1, R0

END

OUTPUT:

R0=0X000000FF

R1=0X000000CD

R2=0X000001CC

AREA MULTIPLY, CODE, READONLY

ENTRY

MOV R0,#0X000000FF

MOV R1,#0X000000FF

MUL R2, R1, R0

END

OUTPUT:

R0=0X000000FF

R1=0X000000CD

R=0X0000FE01

AREA SUBTRACT, CODE, READONLY

ENTRY

MOV R0,#0X000000FF

MOV R1,#0X000000CD

SUB R2, R1, R0

END

OUTPUT:

R0=0X000000FF

R1=0X000000FF

R2=0X00000000

-----------------------------------------------------------------------------------------

ALP TO ADD 10 NUMBERS

AREA ADDITION, CODE, READONLY

start MOV R5,#10

MOV R0,#0

MOV R1,#1

loop ADD R0, R0, R1

SUB R5, R5, #1

CMP R5, #0

BNE LOOP

LDR R4,=RESULT

STR R0,[R4]

XSS B XSS

AREA data2, data, readwrite

Result DCD 0X0

END

OUTPUT:

R0=0X00000037

R1=0X0000000B

R4=0X40000000

R5=0X00000000

-----------------------------------------------------------------------------------------

ALP TO ADD ARRAY OF 16-BIT NUMBER FROM ARRAY AND STORE IN 32-BIT

AREA ADDITION, CODE, READONLY

start MOV R5,#3

MOV R0,#0

LDR R1,=VALUE1

LOOP LDR R2,[R1],#2

LDR R3, MASK

AND R2, R2,R3

ADD RO, RO,R2

SUBS R5, R5,#2

CMP R5,#0

BNE LOOP

LDR R4,=RESULT

STR R0,[R4]

XSS B XSS

MASK DCD 0X0000FFFF

VALUE1 DCW 0X1111, 0X2222, 0X4444

AREA DATA2, DATA, READWRITE

RESULT DCD 0X0

END

OUTPUT:

R0=0X00007777

R1=0X0000003E

R20X00004444

R3=0X0000FFFF

R4=0X40000000

Program 3:

AREA ADDITION,CODE,READONLY

START

MOV R5,#3

MOV R0,#0

LDR R1,=VALUE1

LDR R3, MASK

AND R2,R2,R3

AND R0,R0,R2

SUBS R5,R5 #2

CMP R5 #0

BEM LOOP

LDR R4 = RESULT

STR R0,[R4]

XSS BXSS

Mask DCW 0x1111,0x2222,0x4444

AREA DATA1 DATA READWRITE

RESULT DCD 0X0000

[\\result](file:///\\result)

before execution

R0=0x00000000

R1=0x00000000

R2=0x00000000

R3=0x00000000

R4=0x00000000

after execution

R0=0x00007777

R1=0x0000003E

END

**Program 4:**

AREA FINB,CODE,READONLY

MOV r0,#10;number of fibonocci numbers to generate

MOV r1,#0;f0=0

MOV r2,#1;f1=1

LDR r5,=result\_array

STR r1,[r5],#4;store f0

STR r2,[r5],#4; store f1

LOOP

ADD r3,r1,r2;next Fibonacci number

STR r3,[r5]#;store it

MOV r1,r2;update r1

Mov r2,r3;update r2

SUBS r0,r0,#1;decrement counter

CMP r0,#0

BNF LOOP;repear if not zero

Done b done

AREA DATA1,DATA,READWRITE

Result\_array DCD 6X000

END

//result

Before execution

R0=0X00000000

R1=0X00000000

R2=0X00000000

R3=0X00000000

After execution

R0=0X00000000

R1=0X00000037

R2=0X00000059

R3=0X00000059

Address: 0X40000000: 00000001,00000002,00000002,00000003,00000005,000000008,0000000D,00000015,00000037,00000059

**Program 5:**

AREA SQUARE,CODE,READONLY

START

LDR r0,=table 1

LDR r1,r1,|5|#0X2

Add r0,r0,r1;load address of element in look up table

LDR r3[r0];get source of given on in r3

RSS BXSS

Table 1 DCD 0X00000000

DCD 0X00000001

DCD 0X00000004

DCD 0X00000009

DCD 0X00000010

DCD 0X00000019

DCD 0X00000024

DCD 0X00000031

DCD 0X00000040

DCD 0X00000051

DCD 0X00000661;

Area

Data1,data,readwrite

Result DCD 0X00000000

End

[\\result](file:///\\result)

Before execution

R0= 0X00000000

R2=0X00000000

After execution

R0=0X0000001A

R2=0X00000002