**ARM PROGRAMS:**

**1) To count +ve, -ve n zeros**

**AREA count,CODE**

**ENTRY**

**adr r0,array ....load address**

**mov r2,#0 ....reference number**

**mov r6,#0x0a ....set counter**

**mov r3,#0 ....for storing +ve num**

**mov r4,#0 .... " " -ve "**

**mov r5,#0 .... " " 0 "**

**mov r7,#0**

**loop ldr r1,[r0,r7] ....load first number**

**cmp r1,r2 ....r2-r1**

**addgt r3,r3,#1 .....add if r1>r2**

**addlt r4,r4,#1 .... " " r1<r2**

**addeq r5,r5,#1 .... " " r1=r1=0**

**add r7,r7,#4**

**subs r6,r6,#1 ....dec counter**

**bne loop**

**stop b stop**

**ALIGN**

**array DCD 1,2,0,1,0,2,0,-1,0,-7**

**end**

1. **To add 2 64 bit num**

**AREA add,CODE**

**ENTRY**

**adr r0,array**

**ldr r1,[r0] ......lower 16bits of 1st number**

**ldr r2,[r0,#4] ......higher 16bits of 1st number**

**ldr r3,[r0,#8] ......lower 16bits of 2nd number**

**ldr r4,[r0,#0x0c] ..... higher 16bits of 2nd number**

**mov r7,#0 ......for carry**

**adds r5,r1,r3 .....s in adds will update cpsr**

**adc r6,r2,r4**

**adc r7,r7,#0 ......for carry**

**stop b stop**

**ALIGN**

**array DCD 0xffffffff,0xffffffff,0xffffffff,0xffffffff**

**end**

1. **Y=AB+CD**

**AREA MULT,CODE**

**ENTRY**

**adr r0,array0**

**ldr r7,[r0]**

**adr r1,array1**

**ldr r8,[r1]**

**adr r2,array2**

**ldr r9,[r2]**

**adr r3,array3**

**ldr r10,[r3]**

**mul r4,r8,r7**

**mul r5,r9,r10**

**add r6,r4,r5**

**stop b stop**

**ALIGN**

**array0 DCD 2**

**array1 DCD 2**

**array2 DCD 2**

**array3 DCD 2**

**END**

1. **LAREST NUMBER**

**AREA trans,CODE**

**ENTRY**

**adr r0,array**

**mov r1,#0 …ref num**

**mov r3,#0x0a …counter**

**loop ldr r4,[r0]**

**cmp r4,r1 r1-r4**

**movhi r1,r4 …..mov if higher(if carry is set)**

**add r0,r0,#4**

**subs r3,r3,#1**

**bne loop**

**stop b stop**

**ALIGN**

**array dcd 0,0,0x37,0x46,0x56,6,7,8,9,0x86**

**END**

1. **SMALLEST NUMBER**

**AREA trans,CODE**

**ENTRY**

**adr r0,array**

**mov r1,#0xff**

**mov r3,#0x0a**

**loop ldr r4,[r0]**

**cmp r1,r4**

**movhi r1,r4**

**add r0,r0,#4**

**subs r3,r3,#1**

**bne loop**

**stop b stop**

**ALIGN**

**array dcd 2,1,0x37,0x46,0x56,0,7,8,9,0x86**

**END**

1. **HEX TO ASCII**

**AREA ASCII,CODE**

**ENTRY**

**ADR R0,ARRAY**

**LDR R1,[R0]**

**MOV R2,R1**

**AND R2,#0x0F**

**MOV R3,R1,LSR #04**

**CMP R2,#09**

**ADDLS R4,R2,#0x30**

**ADDHI R4,R2,#0x37**

**CMP R3,#09**

**ADDLS R5,R3,#0x30**

**ADDHI R5,R3,#0x37**

**STOP B STOP**

**ALIGN**

**ARRAY DCD 0x5A**

**END**

1. **UPACKED BCD TO BINARY**

**AREA ASCII,CODE**

**ENTRY**

**ADR R0,ARRAY**

**ADR R1,ARRAY1**

**LDR R2,[R0]**

**LDR R3,[R1]**

**MOV R3,R3,LSL#4**

**ADD R3,R3,R2**

**STOP B STOP**

**ALIGN**

**ARRAY DCD 07**

**ARRAY1 DCD 09**

**END**

**8051 PROGRAMS:**

**FOR EXAM INSTEAD OF “HERE:SJMP HERE” AND “END” WRITE ONLY “RET”.**

1. **COUNT ODD N EVEN**

**MOV DPTR,#2500H**

**MOV R0,#0AH …COUNTER**

**MOV R1,#00H ….EVEN NUM**

**MOV R2,#00H ….ODD NUM**

**CLR C**

**L1: MOVX A,@DPTR**

**RRC A**

**JNC L2**

**INC R2**

**SJMP K0**

**L2: INC R1**

**K0: INC DPTR**

**DJNZ R0,L1**

**MOV A,R1**

**MOVX @DPTR,A**

**INC DPTR**

**MOV A,R2**

**MOVX @DPTR,A**

**HERE: SJMP HERE**

**END**

1. **+VE/-VE SERIES**

**MOV DPTR,#2500H**

**MOV R0,#0AH**

**MOV R1,#00H ......EVEN NUM**

**MOV R2,#00H ......ODD NUM**

**CLR C**

**L1: MOVX A,@DPTR**

**RLC A ......ROTATE LEFT THROUGH CARRY**

**JNC L2**

**INC R2**

**SJMP K0**

**L2: INC R1**

**K0: INC DPTR**

**DJNZ R0,L1**

**MOV A,R1**

**MOVX @DPTR,A**

**INC DPTR**

**MOV A,R2**

**MOVX @DPTR,A**

**HERE: SJMP HERE**

**END**

1. **TO COUNT NUMBER OF 1’S IN A GIVEN BYTE**

**MOV DPTR,#2500H**

**MOV R0,#08H**

**MOV R1,#00H**

**CLR C**

**MOVX A,@DPTR**

**L1: RLC A**

**JNC L2**

**INC R1**

**L2: DJNZ R0,L1**

**INC DPTR**

**MOV A,R1**

**MOVX @DPTR,A**

**HERE: SJMP HERE**

**END**

1. **X^2+X+1**

**MOV DPTR,#2500H**

**MOVX A,@DPTR**

**MOV B,A**

**MOV R1,A**

**MUL AB …….X^2**

**ADD A,#01H …….SUM OF LOWER BYTE OF X^2 AND 1**

**ADDC A,R1 .. SUM OF LOWER BYTE OF X^2 , X ,1 AND CARRY**

**INC DPTR**

**MOVX @DPTR,A**

**MOV A,B**

**ADDC A,#00H**

**INC DPTR**

**MOVX @DPTR,A**

**HERE: SJMP HERE**

**END**

1. **SUM N AVERAGE**

**MOV DPTR,#2500H**

**MOV R0,#0AH**

**MOV R1,#00H**

**L1: MOVX A,@DPTR**

**ADD A,R1**

**MOV R1,A**

**INC DPTR**

**DJNZ R0,L1**

**MOVX @DPTR,A**

**MOV B,#0AH**

**DIV AB**

**INC DPTR**

**MOVX @DPTR,A**

**MOV A,B**

**INC DPTR**

**MOVX @DPTR,A**

**HERE: SJMP HERE**

**END**

1. **2 DIGIT DECIMAL TO HEX**

**MOV DPTR,#2500H**

**MOVX A,@DPTR**

**MOV R1,A**

**ANL A,#0FH**

**MOV R2,A**

**MOV A,R1**

**ANL A,#0F0H**

**MOV R3,#04H**

**L1: RR A**

**DJNZ R3,L1**

**MOV B,#0AH**

**MUL AB**

**ADD A,R2**

**INC DPTR**

**MOVX @DPTR,A**

**HERE: SJMP HERE**

**END**

1. **HEX TO DECIMAL**

**MOV DPTR,#2500H**

**MOVX A,@DPTR**

**MOV R1,A**

**MOV B,#64H**

**DIV AB**

**INC DPTR**

**MOVX @DPTR,A**

**MOV A,B**

**MOV B,#0AH**

**DIV AB**

**MOV R3,#04H**

**L1: RL A**

**DJNZ R3,L1**

**ADD A,B**

**INC DPTR**

**MOVX @DPTR,A**

**HERE: SJMP HERE**

**END**

1. **2 DIGIT HEX TO ASCII**

**MOV DPTR,#2500H**

**MOVX A,@DPTR**

**MOV R6,A**

**ANL A,#0FH**

**MOV R1,A**

**ACALL asc**

**MOV A,R6**

**ANL A,#0F0H**

**MOV R3,#04H**

**L4: RR A**

**DJNZ R3,L4**

**MOV R1,A**

**ACALL asc**

**ACALL here**

**asc: CJNE R1,#09H,L1**

**ADD A,#30H**

**SJMP L3**

**L1: JC L2**

**ADD A,#37H**

**SJMP L3**

**L2: ADD A,#30H**

**L3: INC DPTR**

**MOVX @DPTR,A**

**ret**

**here: sjmp here**

**end**

1. **TO EXCHANGE FROM INTERNAL TO EXTERNAL**

**MOV DPTR,#2500H**

**MOV R1,#20H**

**MOV R2,#0AH**

**L1: MOVX A,@DPTR**

**XCH A,@R1**

**MOVX @DPTR,A**

**INC DPTR**

**INC R1**

**DJNZ R2,L1**

**here: sjmp here**

**end**

1. **EXCHANGE INTERNAL TO INTERNAL**

**MOV R0,#20H**

**MOV R1,#30H**

**MOV R2,#0AH**

**L1: MOV A,@R0**

**XCH A,@R1**

**MOV @R0,A**

**INC R0**

**INC R1**

**DJNZ R2,L1**

**here: sjmp here**

**end**

1. **LARGEST NUMBER**

**MOV DPTR,#2500H**

**MOV R1,#00H ....REF NUMBER**

**MOV R2,#0AH ....COUNTER**

**L1: MOVX A,@DPTR**

**CJNE A,01H,L2 ....A-R1(address of R1 is 01h)**

**L2: JNC L3**

**SJMP L4**

**L3: MOV R1,A**

**L4: INC DPTR**

**DJNZ R2,L1**

**MOV A,R1**

**MOVX @DPTR,A**

**here: sjmp here**

**end**

1. **SMALLEST NUMBER**

**MOV DPTR,#2500H**

**MOV R1,#0FFH**

**MOV R2,#0AH**

**L1: MOVX A,@DPTR**

**CJNE A,01H,L2**

**L2: JC L3**

**SJMP L4**

**L3: MOV R1,A**

**L4: INC DPTR**

**DJNZ R2,L1**

**MOV A,R1**

**MOVX @DPTR,A**

**here: sjmp here**

**end**

1. **ADDITI0N OF FIRST 10 EVEN NOS**

**MOV DPTR,#2500H**

**MOV R0,#0AH**

**MOV A,#00H**

**MOV R1,#00H**

**L1: ADD A,R1**

**INC R1**

**INC R1**

**DJNZ R0,L1**

**MOVX @DPTR,A**

**here: sjmp here**

**end**