	write an assembly language program to implement
	division by repetitive subtraction.
	Scotton Letter
	AREA RESET, DATA, READONLY
	EXPORTVectors
	Vectors
	DCD 0x10001000
	DCD Reset-Handley
	Rest Hardler moodoolko engopoolko en
	LDR RO = DESTINO
	LDR RI, = DEST2
	Mov R2,#0
	LDR R3,=QUO
	LDR RS FREM
	LDR R5, [RO]
	LOR RG, [RI]
	LOOP SUB RS, RS, RS, RG
	ADD R2, R2, #1
	CMP R5, R6
	BHS LOOP
	STR R2,[R3]
	STR RS,[R4]
	STOP B STD8
	DESTI DCD 46
	DEST2 DED 9
	AREA data, DATA, READWRITE
	QUE DED
1	REM DCD 0
_	END

ED	ES
COTPUT	
First output Membel 0x100000000: 05 00 00 0100 00 Regulary 0x2(21 → 0x30 → 6x30 22 → 0x00 23 → 0x10000000 → 0x1000000 24 → 0x10000000 → 0x1000000 25 → 0x2 F → 0x000000 Regulary 0x00000000 0x00000000000000000000000	
<u>-</u> - <u>-</u>	
	4 = 1

	Find the Su	ion of n ratu	xu) oumber	м.	
(2)		RESET, DATA			
		Vectors		1000	-
		- 0 X MC () =		0826	0.9
		10001000			
		et_Handler			
	ALIGIN	70,00	(-	0010	23
	A REA T	ny code, CODE, F	READONLY	0001X0	FA
	ENTRY	· 125/20	LENGANLE		
		mywode CODE		Reset_Han	dles
	Reset Handler	C Royale Honda	£3		-
		RO = NUMI	00 00 30	00000	ONO
	LDR F	21, [RD]		*	
		7, = DEST			
	MLA F	21,121,121,121			
		21#2			
	MOV R	3,#0			
	LOOP SUB	RI, RI, R2			
	ADD F	23, R3,#1			
,	(top				
	BHS				
		R3, [R7]			-
	STOP B STOP				
	NUM DCD 5				
/		ata, DATA, RE	ADWRITE		
	DEST DLD O	R.S. #270			
	END.	-61			
1					
	HRO.	1000			
		10.034			
		200 200 500			
-					

				EDGA
	2 - 25.00	hauri bi	edica a do o	selv Bell-ju
0	wpu	Grande	ATAC ISZE	4134
			tood is de	EX Parel
Ro	0×2C		0× 2 (APLICA V
RI	0×1E)	0100	0.10 O.10
R2	0×02)	0x 02	
R3	0 × 0 0	<u>—)</u>	OXOF	5
R7	0×100000	000	0×1000000	
				- Value
m	ewost	Hills A 2	CASTON TO ARTHURST THE	Rear claveler
			115 0	
0 %	- 000000	0F 00 00	0 00 0 15 0	4 841
			12.94 =	Fa 90 L
		-	19 12d 12d, 1	<u> </u>
			<u> </u>	28 V 617
			(2) (2) (3)	
			14.89.8	
	-		19.12	
		-	900	
			[19]. 89	
				SOTA SOTA
			, = =	2 020 must
		-	as attach ou	
				0 00 730
				ana
			Sant La	



u	R	В	Δ	N
Pre-	m	ū	e,	
-		X.	-	~
Birm	ш	^	"	-4

(3)	Write an assembly language problem program to find
	OICD and Lim of two 8 bits.
	AREA RESET, DATA, READONLY
	EXPORT Vectory
	Victor
	DCD 0x10001000
	DCD Reset Handlex
	ALIGIN
	AREA mycode, cope, READONLY
	ENTRY
	EXPORT Reset Handley
	Rest Handler
	LDR RD, = NUM)
	LDRB RI, IRO]
	LDR RO, = NUMZ
	LORB R2, LRO]
	(mp R1, R2
	BHI MS(
	MDY R3, R1
	MOV RI,R2
	MIBYE RZIR3
	MSC MOY R3, R2
	FIX MOV R4, R1
	MIDY RS, R2
	DIV SUBS RY, R4, R3
	(MP R4, #DXD
	BWI BE?
	BHI DIV
	BEG DIV2
/	RES SUB R3, R3#1
	BAL FIX
	DIY2 SUBS RS, RS, RS, RS
	cmp 25, #0x0
	. BODI. RES 2.

	EBBAN
BHI PIV2	on topical photosic as world
BEG FIN	the Power districts have given
V 21MC CH	137, 9176, 1371M A224
RES2 SUB P3, P3,	# 1 40929
BAL FIX	MOLIN
	1,5 80 Fa 900 Kg . 0.10
FIN LDR RS, = GIC	D villagely total and
STR R3, [RS]	Maria
MUL REIRIFE	25, Jan skayer Hand
0x0#) Fg vom	SAIRS
	ALDSON LOAD TARRY
DI SUB RG, RG	Parte Mandan
(mp R6,#0)	XD 10001 = 00 001
ADD R7,R7,	# 1 1341 18 8401
BEQ FINL(M	11944 = 68 20 .
BHI DI	1081, 18 201
FINLIM LDR R8,=LC STR R7, LR	2
	May PLES
STOP B STOP	May Priks
NUMI DED DX4	MSC MOX RAJES
NUM2 DCD 0x2	FIX MOV FH. EL
AREA data, DATA	, READWRITE
GICD DED O	DIV SUBS BYDY P3
Lew Dep 0	0x0 (+ + + + + m)
END	229 7008
2 Output	AIG THE
	. ded pivi
R\$: 0x1000000	0 - 02000000
R#: 0x \$0000004	P → 04 00 00 00
£0→0×00→0×6($R3 \rightarrow 0\times00 \rightarrow 0\times02$
RI - 1 0 x 00 -> 0 x 0 4	RT -> OXOU -> OXUY
R2.→ 0×00. → 0×02	. KZ-10× 10000000
	R6-> 0x 10000004

u.	R.	в	A	No.
_	r	ø	•	mg.
_		x.	-	м.
-	_	•	~	wii .

	[103]
4)	WAP to convent 2 digit hexadecimal to ascii
	AREA RESET PATA, READONLY
	EXPORT Vectors
	Vedors Land
	DCD 0x10001000 68x4
	DCD Result Handler
	ALIGN PORMERON TO THE GOOD GOO X A THE DO
	AREA MYWWW, TRODE, READONLY
	ENTRY
	EXPORT Revel Handlet
	Reset Hander 1999 1998 00 00 00 00
	LOR RO, = NUMI
	LDR RI, [RO]
	AND RB, RI, #DXOF
	MOV R3, R1, LSR, #4
	cmp R2, #0x D9
	BLS DOWN
	Add R2, R2, # 0×09
	DOWN ADD RZIRZ, #0x30
	second (mp R3, #0×09
	BLS Down 2
	Add R3, R3, #0x07
	Down 2 ADD R3, R3, # 0x30
	STORE LOR R4, = DEST 6
	STE RZ,[R4]
	LDR RS, = DESTIB
	STR R3, [R5]
	STOR B STOP
	ANUMI DOD OX3A
	AREA data, DATA, READWRITE
	DEST O PCD O
	DESTIO DCDO
	END

	EDG	
(3)	Output	7.8W
	RO-) 0x00000000 0x38	AREA
	RI → 0×00000080 → 0×3A	
	$R_2 \rightarrow 0 \times 00000000 \rightarrow 0 \times 10$	Relax V
	R3 -> 0x0000000-> 0x33	999
	R4 -> 0x00000000 -> 0x10000000	429
	RS-10x00000000 -10x100000004	DIA
	My Market	
		ATMI
	0x 100000000 indicate 1002 [2	
	4,0000 00 33 000000	
	Francisco de la companya della compa	
	[09] [8	
	D RA RI BOXOF	0
	2 (23 . E) 12E . FH	01.1
	Maria	
	dd 82,82, # 0x09	
	ND ROLL CON SO	
	(vnp p3, d0x0d	
	RIS Down 2	
	add K3. R3. #0x07	
	ROD R3, R8, dt ox3g at ro	
	LOK SH, = DEST a	3-201
	CHAJISA ATA	
	01T23G = ,28 AQJ	
	[Zalen ata	
	9072 8	201
	PER STATE OF THE SERVE COOP DIS	
	LE JOHN DATE, READURITE	
	TO THE RESIDENCE TO THE PARTY OF THE PARTY O	728
	The second secon	DITESTIN
	S	

u	R	в	Α	м
- pa	i jiri	10	c	÷
- 10-	-	v	_	-

	5)	WAP to convert 32 bit BCD number in propacked to
EXPORT VECTORS VECTORS DED OXIODO 1000 OLD RESEL HANDLE ALT GIN AREA rayLocke, CODE IREADONLY ENTRY EXPORT RESEL HANDLES ROSEL HANDLES ROSEL HANDLES LDR RO, = NUM1 LDR RI, [Ro] LDR RO, = NUM2 LDR RO, = NUM2 LDR RS, LSL # 4 MODS R4, RI, R3 LDR RS, = DEST STR R4, [RS] STOP B STOP NUM1 OCD OXI NUM2 DCO OX2 AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 RI: 0x00000000 → 0x01 RA: 0x00000000 → 0x01 PA: 0x00000000 → 0x20 R4: 0x00 → 0x21	71	packed forms
EXPORT VECTORS VECTORS DED OXIODO 1000 OLD RESEL HANDLE ALT GIN AREA rayLocke, CODE IREADONLY ENTRY EXPORT RESEL HANDLES ROSEL HANDLES ROSEL HANDLES LDR RO, = NUM1 LDR RI, [Ro] LDR RO, = NUM2 LDR RO, = NUM2 LDR RS, LSL # 4 MODS R4, RI, R3 LDR RS, = DEST STR R4, [RS] STOP B STOP NUM1 OCD OXI NUM2 DCO OX2 AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 RI: 0x00000000 → 0x01 RA: 0x00000000 → 0x01 PA: 0x00000000 → 0x20 R4: 0x00 → 0x21	\dashv	A CONTRACT DATA DESCRIPTIV
DCD OXIGODIDO O OCD Reset Handle ALTIGN AREA raycode, CODE IREADONLY ENTRY EXPORT Reset Handles Reset Handles LDR RO, = NUM1 LDR RI, [Ro] LDR R2, [Ro] MOV R3, R2, LSL # 4 ADDS R4, R1, R3 LDR RS, = DEST STR R4, [RS] STOP B STOP NUM1 OCD OX1 NUM2 PCO OX2 AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 R1: 0x00000000 → 0x01 R3: 0x00 → 0x21	_	
DCD ESSET HUNDER DCD RESET HUNDER ALIGN AREA MYCOCLE, CODE IREADONLY ENTRY EXPORT RESET HANDES RESET HUNDER LDR RO, = NUMP LDR RI, [Ro] LDR R2, [Ro] MOY R3, R2, LSL # 4 ADDS R4, R1, R3 LDR RS, = DEST STR R4, [RS] NUM1 OCD OXI NUM2 DCD OX2 AREA dota, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 R1: 0x00000000 → 0x01 P4: 0x00 → 0x21	\dashv	
DCD Reset Harder ALT GIN AREA MYCOCLE, CODE IREADONLY ENTRY EXPORT Reset Harder Reset Harder LDR RO, = NUMP LDR RI, [Ro] LDR R2, [Ro] MOV RB, R2, LSL # 4 ADDS R4, R1, R3 LDR RS, = DEST STR RY, [RS] STOP B STOP NUMI OCD OXI NUM2 PCD OX2 AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 RR: 0x00000000 → 0x01 PA: 0x00 → 0x21	\dashv	
ALT GN AREA MYCOCLE, CODE READONLY ENTRY EXPORT Reset Hardes Reset Hardes LDR RO, = NUMI LDR RO, = NUMI LDR RO, = NUMI LDR RO, = NUMI ADDS RY, ERD MOV RB, RZ, LSL # 4 ADDS RY, RI, R3 LDR RS, = DEST STR RY, ERS NUMI DCD DXI NUMI DCD DXI NUMI DCD DXI NUMI DCD DXI PETT DCD D END Output: RO: 0x00000000 → 0x20 R8: 0x00000000 → 0x20 R8: 0x00000000 → 0x20 P4: 0x00 → 0x21	_	The same of the sa
ENTRY EXPORT Rest Hardes Rest Hardes LDR RO, = NUM! LDR RO, = NUM! LDR RO, = NUMP LDR RO, = NUMP LDR RO, = NUMP LDR RO, = NUMP MOV RB, R2, LSL # 4 ADDS R4, R1, R3 LDR RS, = DEST STR R4, LRS] STOP B STOP NUMI OCD DX! NUM2 DCD DX2 AREA data, DATA, READWRITE DEST DCD D END OULPUT: RO: 0x00000000 → 0x20 R8: 0x00000000 → 0x01 R8: 0x00000000 → 0x20 P4: 0x00 → 0x21	-	
ENTRY EXPORT Resal Hardes Resal Hardes LDR RO, = NUMM LDR RI, [Ro] LDR RO, = NUMM LDR R2, [RD] MOV R3, R2, LSL # 4 ADDS R4, R1, R3 LDR RS, = DEST STR R4, [RS] STOP B STOP NUMI OCD OXI NUM2 DCD OX2 AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 R1: 0x00000000 → 0x01 RM: 0x00000000 → 0x01 RM: 0x00000000 → 0x01 RM: 0x00000000 → 0x20 R4: 0x00 → 0x21	-	
EXPORT Reset-Handle Reset Handle LDR RO, = NUMM! LDR RI, [Re] LDR RO, = NUMM2 LDR RO, = NUMM2 LDR RO, = NUMM2 MOV RB, R2, LSL # 4 ADDS R4, R1, R3 LDR RS, = DEST STR R4, [RS] STOP B STOP NUMI DCD DX! NUM2 DCD DX2 AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 RA: 0x00000000 → 0x01 RA: 0x00000000 → 0x02 RA: 0x00 → 0x21	-	·
Reset Hardes LDR RO, = NUM! LDR RI, [Ro] LDR Ro, = NUMP LDR RO, = NUMP LDR RO, = NUMP LDR RO, = NUMP MOV RB, R2, LSL # 4 ADDS R4, RI, R3 LDR RS, = DEST STR R4, [RS] STOP B STOP NUMI DCD OX! NUM2 DCD OX AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 R1: 0x00000000 → 0x01 RA: 0x00000000 → 0x20 P4: 0x00 → 0x21	+	
LDR RO, = NUMN LDR RI, [Ro] LDR Ro, = NUMN LDR R2, [Ro] MOV R3, R2, LSL # 4 ADDS R4, RI, R3 LDR PS, = DEST STR R4, [RS] NUMI OCD OXI NUM2 PCD OXI AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 R1: 0x00000000 → 0x01 RA: 0x00000000 → 0x20 PA: 0x00 → 0x21	-	
LDR RI, [Ro] LDR RO, = NUME LDR R2, [Ro] MOV R3, R2, LSL # 4 ADDS R4, RI, R3 LDR RS, = DEST STR R4, [RS] STOP B STOP NUMI DCD DXI NUM2 DCD DX AREA data, DATA, READWRITE DEST DCD b END Output: RO: 0x00000000 → 0x20 R1: 0x00000000 → 0x01 R2: 0x00000000 → 0x20 P4: 0x00 → 0x21	+	
LDR RO, = NUMP LOR R2, [Ro] MOV R3, R2, LSL # 4 ADDS R4, R1, R3 LDR RS, = DEST STR R4, [RS] STOP B STOP NUM1 OCD OX1 NUM2 DCD OX2 AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 R1: 0x00000000 → 0x01 PA: 0x00 → 0x21	-	
LOR R2, [R0] Mov R3, R2, LSL # 4 MODS R4, R1, R3 LDR RS, = DEST STR R4, [RS] STOP B STOP NUM1 OCD OX1 NUM2 PCD OX2 AREA data, DATA, READWRITE DEST DCD D END Output: R0: 0x00000000 → 0x20 R1: 0x00000000 → 0x01 R2: 0x00 → 0x21 P4: 0x00 → 0x21	-	
MOV RB, R2, LSL # 4 ADDS R4, R1, R3 LDR RS, = DEST STR R4, [RS] STOP B STOP NUMI DCD DXI NUM2 DCD DX2 AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 R1: 0x00000000 → 0x01 RA: 0x00000000 → 0x20 P4: 0x00 → 0x21	-	
ADDS R4, R1, R3 LDR RS, = DEST STR R4, [RS] STOP B STOP NUM1 DCD OX1 NUM2 DCD OX2 AREA data, DATA, READWRITE DEST DCD D END Output: R0: 0x00000000 → 0x20 R1: 0x00000000 → 0x01 PM: 0x00000000 → 0x20 P4: 0x00 → 0x21		LDR R2, [RD]
LDR RS, = DEST STR RY, [RS] STOP B STOP NUMI DCD DXI NUM2 PCD DX2 AREA data, DATA, READWRITE DEST DCD D END OUTPUT: RO: 0x00000000 → 0x20 R1: 0x00000000 → 0x01 RA: 0x00000000 → 0x20 P4: 0x00 → 0x21	_	
STR RY, [RS] STOP B STOP NUMI DCD 0XI NUM2 DCD 0X2 AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 RI: 0x00000000 → 0x01 RA: 0x00000000 → 0x20 PA: 0x00 → 0x21		
STOP B STOP NUMI OCD 0XI NUM2 PCD 0X2 AREA data, DATA, READWRITE DEST DCD D END OUTPUT: RO: 0x00000000 → 0x20 RI: 0x00000000 → 0x01 RA: 0x00000000 → 0x20 PA: 0x00 → 0x21	-	
NUMI DCD 0XI NUM2 DCD 0X2 AREA data, DATA, READWRITE DEST DCD D END OULPUT: RO: 0x00000000 → 0x20 RI: 0x00000000 → 0x01 RA: 0x00000000 → 0x20 P4: 0x00 → 0x21	-	
NUM2 PCD 0X2 AREA data, DATA, READWRITE DEST DCD D END Output: RO: 0x00000000 → 0x20 RI: 0x00000000 → 0x01 RA: 0x0000000 → 0x02 PA: 0x00 → 0x20 P4: 0x00 → 0x21		
AREA data, DATA, READWRITE DEST DID D END Output: RO: 0×000000000 → 0×20 RI: 0×00000000 → 0×01 RA: 0×00000000 → 0×02 PA: 0×00 → 0×20 P4: 0×00 → 0×21	-	
DEST D(D D END Output: R0: $0 \times 000000000 \rightarrow 0 \times 20$ R1: $0 \times 000000000 \rightarrow 0 \times 01$ R2: $0 \times 000000000 \rightarrow 0 \times 02$ P4: $0 \times 00 \rightarrow 0 \times 20$	-	
END Output: RO: $0 \times 000000000 \rightarrow 0000$ R1: $0 \times 000000000 \rightarrow 0000$ R2: $0 \times 000000000 \rightarrow 0000$ P3: $0 \times 000 \rightarrow 0000$ P4: $0 \times 00 \rightarrow 0000$	-	
Output: R0: $0 \times 000000000 \rightarrow 0000$ R1: $0 \times 000000000 \rightarrow 0000$ R2: $0 \times 000000000 \rightarrow 0000$ P3: $0 \times 000 \rightarrow 0000$ P4: $0 \times 00 \rightarrow 0000$	-	
$R1: 0 \times 00000000 \longrightarrow 0 \times 01$ $R2: 0 \times 000000000 \longrightarrow 0 \times 02$ $R3: 0 \times 00 \longrightarrow 0 \times 20$ $R4: 0 \times 00 \longrightarrow 0 \times 21$	4	END STR PT
$\begin{array}{c} RA: 0 \times 00000000 \longrightarrow 0 \times 02 \\ RA: 0 \times 00 \longrightarrow 0 \times 20 \\ RA: 0 \times 00 \longrightarrow 0 \times 21 \end{array}$		Output: RO: 0x00000000 -> 0x20
23: 0x00→ 0x20 P4: 0x00→ 0x21		$R1: 0 \times 0 0 0 0 0 0 0 0 \longrightarrow 0 \times 0$
P4: 0×00 → 6×21		RA: 0x 000000000 > 0x 02
		23: 0x00-) 0x20
125: 0x 1000000		P4: 0×00 → 0×21
		125:0x1000000