

7. Se dă un șir de dublucuvinte. Să se obțină șirul format din octeții superiori ai cuvintelor superioare din elementele șirului de dublucuvinte care sunt divizibili cu 3.

Metoda 1 (first->last)

Data	Code
<pre> S DD 12345678h, 1A2B3C4Dh, 0FE98DC76h, 03111111h, 19010203h, 3FA0A0A0h S_Len EQU (\$-S)/4 ; = 6 D TIMES S_Len RESB 1 </pre>	<pre> mov ESI, S ; ESI = S = source mov EDI, D ; EDI = D = destination cld ; DF = 0 (ESI/EDI+) mov DL, 3 ; DL = 3 mov ECX, S_Len ; ECX = S_Len = 6 iterate: lodsd ; EAX = S[ESI] = X1_X2_X3_X4h, ESI+=4 push CX ; backup counter reg. mov CL, 8 ; CL = 8 rol EAX, CL ; EAX = X2_X3_X4_X1h pop CX ; restore counter reg. mov BL, AL ; BL = AL = X1h (backup AL) mov AH, 0 ; AH = 0 <=> AX = AL div DL ; AL = AX/3, AH = AX%3 or AH, AH ; AH =AH, ZF=1 if AH == 0; (cmp AH,0) loopnz iterate; jp if(!ZF) <=> if(X1%3!=0)continue; ; ECX==0 ; (X1h % 3 == 0) : mov AL, BL ; AL = BL = X1h (restore AL) stosb ; D[EDI++] = AL loop iterate ; if (ECX>0) goto iterate </pre>
Output	
<pre> D DB 12h, 03h, 3Fh </pre>	

Data	Code
<p>S DD 12345678h, 1A2B3C4Dh, 0FE98DC76h, 03111111h, 19010203h, 3FA0A0A0h S_Len EQU (\$-S)/4 ; = 6 D TIMES S_Len RESB 1 D_Len RESB 1 ; actual dest length</p>	<pre> mov ESI, S+4*(S_Len-1) ; ESI = 1st(source) mov EDI, D ; ESI = D = destination mov DL, 3 ; DL = 3 mov ECX, S_Len ; ECX = S_Len = 6 iterate: std ; DF = 1 (ESI-) lodsd ; EAX= S[ESI] = X1_X2_X3_X4h,ESI-=4 push CX ; backup counter mov CL, 8 ; CL = 8 rol EAX, CL ; EAX = X2_X3_X4_X1h pop CX ; restore counter mov BL, AL ; BL = AL = X1h (backup AL) mov AH, 0 ; AH = 0 <=> AX = AL div DL ; AL = AX/3, AH = AX%3 or AH, AH ; AH =AH, set ZF=1 if AH == 0 loopnz iterate ; if(!ZF) <=> if(X1h%3!=0) continue; ; ECX==0 ; (X1h % 3 == 0) : mov AL, BL ; AL = BL = A1h (restore AL) cld ; DF = 0 (EDI+) stosb ; D[EDI++] = AL inc BYTE [D_Len] ; D_Len ++ loop iterate ; if (ECX>0) goto iterate xor ECX, ECX ; ECX = 0 mov CL, [D_Len] ; CL = (ECX ==) D_Len = 3 mov ESI, D ; ESI = D mov EDI, ESI ; EDI = ESI = D add ESI, ECX ; ESI = D + D_Len dec ESI ; ESI = D + D_Len - 1 = 1st(D) shr ECX, 1 ; ECX = D_Len/2 = 1 cld ; DF = 0 (ESI/EDI+) reverseD: mov AL, [EDI] ; AL = *EDI := D[i], i=S_Len/2-ECX movsb ; [EDI]=[ESI], EDI++, ESI++ dec ESI ; ESI -= 1, crt. elem. mov [ESI], AL ; [ESI] := D[D_Len-1-i] = AL dec ESI ; ESI -= 1, prev. elem. loop reverseD ; if (ECX>0) goto reverseD </pre>
Output	
<p>D DB 12h, 03h, 3Fh</p>	

CPU - main thread, module 7_0			Registers (FPU)
00402000	\$ BE 00104000	Mov ESI, OFFSET 00401000	EAX: 00000000
00402005	- BF 1B104000	Mov EDI, OFFSET 00401018	EDX: 00402003 7_0.00402003
0040200A	- FC	Cld	ECX: 0039303F
0040200E	B2 03	Mov DL, 3	ESP: 0019FF74
004020D0	B9 06000000	Mov ECX, 6	EBP: 0019FF80
00402012	> AD	LdSt Dword Ptr DS:[ESI]	ESI: 00401018 7_0.00401018
00402015	66:51	PUSH CL, 8	EDI: 00401018 7_0.00401018
00402015	B1 08	ROL EAX, CL	EIP: 0040202A 7_0.0040202A
00402017	D3C0	POP CX	C 0 ES 002B 32bit 0(FFFFFFFF)
00402019	66:59	Mov BL, AL	A 1 CS 0023 32bit 0(FFFFFFFF)
0040201B	88C3	Mov AH, 0	O SS 002B 32bit 0(FFFFFFFF)
0040201D	B4 00	DIV DL	Z 1 DS 002B 32bit 0(FFFFFFFF)
0040201F	F5F2	OR AH, AH	C 0 FS 0053 32bit 0(FFFFFFFF)
00402021	08E4	JNZ SHORT 00402028	T 0 GS 002B 32bit 0(FFFFFFFF)
00402023	✓ 75 03	Mov AL, BL	D 0
00402025	88D8	STOS Byte PTR ES:[EDI]	O 0 LastErr 000000BB ERROR_SEM_NOT_F
00402027	✓ AA	LOOP SHORT 00402012	EFL 00000246 (NO,NB,E,BE,NS,P,FE,LE)
00402029	✗ 6A 00	PUSH 0	ST0 empty 0.0
00402032	L FF15 3C304000	CALL DWORD PTR DS:[<msvcrt.exit>] status HSUCR	ST1 empty 0.0
00402032	0000	Add BYTE PTR DS:[EAX], AL	ST2 empty 0.0
00402034	0000	Add BYTE PTR DS:[EAX], AL	ST3 empty 0.0
00402036	0000	Add BYTE PTR DS:[EAX], AL	ST4 empty 0.0
00402038	0000	Add BYTE PTR DS:[EAX], AL	ST5 empty 0.0
0040203A	0000	Add BYTE PTR DS:[EAX], AL	ST6 empty 0.0
0040203C	0000	Add BYTE PTR DS:[EAX], AL	ST7 empty 0.0
0040203E	0000	Add BYTE PTR DS:[EAX], AL	FST 0000 Cond 3 2 1 0 ESP U O
00402040	0000	Add BYTE PTR DS:[EAX], AL	FCW 027F Prec NEAR, 53 Err 0 0 0 0
00402042	0000	Add BYTE PTR DS:[EAX], AL	Last cnnd 0000:00000000 Mask 1 1 1
00402044	0000	Add BYTE PTR DS:[EAX], AL	XMM0 00000000 00000000 00000000 000000
00402046	0000	Add BYTE PTR DS:[EAX], AL	XMM1 00000000 00000000 00000000 000000
00402048	0000	Add BYTE PTR DS:[EAX], AL	XMM2 00000000 00000000 00000000 000000
0040204A	0000	Add BYTE PTR DS:[EAX], AL	
0040204C	0000	Add BYTE PTR DS:[EAX], AL	
0040204E	0000	Add BYTE PTR DS:[EAX], AL	
00402050	0000	Add BYTE PTR DS:[EAX], AL	
Stack [0019FF70]=00010000 (decimal 65536.)			
Address	Hex dump	ASCII	
00401000	78 56 34 12 40 AC 2B 1A 76 DC 98 FE 11 11 11 03	wU4nH<+~m\$44	0019FF74 75EF6359 Venru RETURN to KERN
00401010	00 00 01 19 00 A0 00 7F 12 03 F8 00 00 00 00	•0!444?42	0019FF7C 75EF6340 @onu KERNEL32.BaseT
00401020	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FF80 7019FD0C = +
00401030	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FF84 7AF7C14 03 RETURN to ntdll
00401040	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FF88 0039303F
00401050	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FF8C DA897AC2 tZr
00401060	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FF90 00000000
00401070	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FF94 00000000
00401080	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FF98 00393030 09
00401090	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FF9C 00000000
004010A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FFA0 00000000
004010B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FFA4 00000000
004010C0	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FFA8 00000000
004010D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00		0019FFAC 00000000

CPU - main thread, module 7_1

```

00402000 < BE 14104000 MOV ESI, OFFSET 00401014
00402005 < BF 18104000 MOV EDI, OFFSET 00401018
0040200A < B2 03 MOV DL, 3
0040200C < B9 06000000 MOV ECX, 6
00402011 > FD STD
00402012 < AD LODS DWORD PTR DS:[ESI]
00402013 < 74 PUSH CX
00402015 < B1 08 MOV CL, 8
00402017 < D3C0 ROL EAX, CL
00402019 < 66:59 POP CX
0040201B < 88C3 MOV BL, AL
0040201D < B4 00 MOV AH, 0
0040201F < F6F2 DIV DL
00402021 < 0854 OR AH, AH
00402023 < JNZ SHORT 0040202F
00402025 < 88D8 MOV AL, BL
00402027 < FC CLD
00402028 < AR STOS BYTE PTR ES:[EDI]
00402029 < INC BYTE PTR DS:[401030]
0040202B < E095 30104000 LOOP SHORT 00402011
00402031 < 31C9 XOR ECX, ECX
00402033 < 8A0D 30104000 MOV CL, BYTE PTR DS:[401030]
00402035 < BE 18104000 MOV ESI, OFFSET 00401018
00402037 < 89F7 MOV EDI, ESI
00402039 < 01CE ADD ESI, ECX
00402042 < 4E DEC ESI
00402043 < D1E9 SHR ECX, 1
00402045 > FC CLD
00402046 < 8A07 MOV AL, BYTE PTR DS:[EDI]
00402048 < A4 MOVS BYTE PTR ES:[EDI], BYTE PTR DS:[ESI]
00402049 < DEC ESI
0040204A < 8B06 MOV BYTE PTR DS:[ESI], AL
0040204C < 4E DEC ESI
0040204D < E2 F6 LOOP SHORT 00402045
0040204F < 6A 00 PUSH 0
00402051 < FF15 3C304000 CALL DWORD PTR DS:[<msvcrt.ex!t>]

```

Stack (0019FF7F)=00010000 (decimal 65536.)

Registers (FPU)

```

EAX 00000000
ECX 00402003 7_1.00402003
EDX 00363012
ESP 0019FF74
EBP 0019FF80
ESI 00401019 7_1.00401019
EDI 00401019 7_1.00401019
EIP 0040204F 7_1.0040204F
C 1 ES 002B 32bit 0(FFFFFFFF)
P 0 CS 0023 32bit 0(FFFFFFFF)
A 0 SS 002B 32bit 0(FFFFFFFF)
7 0 DS 002B 32bit 0(FFFFFFFF)
C 0 FS 0053 32bit 366000(FFF)
T 0 GS 002B 32bit 0(FFFFFFFF)
D 0
0 0 LastErr 000000BB ERROR_SEM_NOTIFY
EFL 00000203 (NO,B,E,BE,NS,P,0,GE,G)
ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
FST 0000 Cond 0 0 0 ES P U 0
FCW 027F Prec NEAR,53 Mask 1 1 1
Last cnd 0000:00000000

```

XMM0 00000000 00000000 00000000 00000000
XMM1 00000000 00000000 00000000 00000000
XMM2 00000000 00000000 00000000 00000000

Address Hex dump

```

00401000 78 5E 34 12 40 3C 2B 1A 7F DC 98 FE 11 11 11 03
00401010 03 02 01 19 0A 00 00 00 00 00 00 00 00 00 00
00401020 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401030 03 00 00 00 00 00 00 00 00 00 00 00 00 00
00401040 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401050 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401060 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401070 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401080 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00401090 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00
004010D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

ASCII

```

wU4nK+...
00401010 00401010
00401020 00401020
00401030 00401030
00401040 00401040
00401050 00401050
00401060 00401060
00401070 00401070
00401080 00401080
00401090 00401090
004010A0 004010A0
004010B0 004010B0
004010C0 004010C0
004010D0 004010D0

```

0019FF74 75EF6359 Venu RETURN to KERN

0019FF7C 75EF6340 @onu KERNEL32.BaseT

0019FF80 0019FFDC 0019FF94 77C7C14 14w RETURN to ntdll

0019FF88 0019FF88 0019FF8C C46D37AD 47m

0019FF90 00000000

0019FF94 00000000

0019FF98 00363000 06

0019FF9C 00000000

0019FFA0 00000000

0019FFA4 00000000

0019FFA8 00000000

0019FFAC 00000000