COAL			
Assignment #2 Name: Ritesh Kumay Roll nos 21K-3961 Sec: BSCCS)-31E			
RESULT is not deterdeclared so MOU RESULT, EAX will give an exxox.			
Before "ret" statement, pop esi sho be there i.e. POP ESI			
ret Addition ENDA			
(i) 11500000 ← 00001 FECh 5 ← 00001 FF0h 6 ← 0000 1 FF4h			
<pre> <pre> <pre> </pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>			

	Question #2:
and a series of the series of	
ma como de Pares	00100101
	00001001
tulida aliku u pagana kerca	00000001 line 2 ZF=0
d.	00100100
	00001001
-	00000000 line 4 Zfz1
ş -1	
	Question #3:
	include ixvine32.inc
-	o Octo.
	axx1 Sdword 40, -90, -67, 98, 78, -45, 0,32
	ass 2 solword 5 DUP (?)
	· code
	main proc
	mov esi, offset arr1
	mov ecx, length of arr1
	pushad
	'call positive values
	popad
	exit
	main endp
- vandation	

	7 (K- 2101
Question # 1:	
	and the second s
include irvine 32. in	
· derta	
A solward 10	
B sdword 5	s inbialized random
N solword 3	Values
-code	
_while: >	-else:
mov eax, N	dec N
cmp eax, 0	(jmp-while
jle endwhile comp eax, 3	- endushile:
cmp eax, 3	exit
je - else	main endp
mou eax, A	end main
cmp N, eax	
in checkB	
sub N, 2	
imp -while:	
checkB:	
mov eax, B	
cmp N, eax	
jng _else	
sub N, 2	
imp_while	The state of the s
	The second section is a second section of the se

Question #5: include irvine 32. inc prompt byte "Enter any number between 0. and 5: ", 0

prompt 2 byte "e", 0

. code · code main proc offset mov edx, sprompt call writestring call readclec ing endel comp al, 5 ini endd BRID BAR cmp al, 1 ine checkfor3 mor edx, offset prompt1 call writestring imp endd checkfor3: ine checkfor2

Charles States V	mov edx, offset prompt1
The second of the second of the	call writestring
igns the training	imp endo
and the second section of the second	checkfor 2:
in the second second	cmp al, 2
Manual manual and	ine checkfor 4
The second secon	mov edx, offset prompt2
· ·	call writestring
	imo enda
1	checkfor 4:
	vocmp al, 4
	tion andel
ti entigras estació, d'és () i regulació d	mov edr, offset prompt2
	call writestring
	endo:
	exit
	main endp
	end main
1	
1	

	2K-3961
Question #6:	
and the second s	3) call writedec
include irvine 32 inc	· Call cost
· data	deci
a dward 100	Jimp -inner
b dword 200	endinner:
c C clustel	deci
i dincol ?	Jimp -outer
i dword ?	endd:
· code	exit exit
The second secon	main endp
main proc	end main
mov i, 5	- Price Price
The state of the s	
cmp c, o	
jng endd mov eax, b	
add eax, a	
mov b, eax	
call writedec	
call cx1f	
p mov i, 5	
-inner:	
ing endiner	
dec a	
add cC, 10	
mov eax, a	
Jan	

14.6

Question #7:	
include ixvine 32-inc	
e chata	
prompt byte "Enter any number: ", space byte " ", o	0
space byte " " 0	the later or the time of the particular particular and the particular particu
·code	
mor edx, offset promp	
call writestring	
call readdec	
mov ebx, eax	
mov ecx, eax	
400 L1:	
push ecx	
(2:	
call writedec	
mor edx, offset space	and the second s
call writestring	
dec eax	in menengan serikan dia diagnositrah nyaétan serikan se
100p (2	
call cx1f	· · · · · · · · · · · · · · · · · · ·
mov eax, ebx	and the second second second second second
pop ecx	
100p 12	
exit	The same of the sa
main endp	
lend main	

the state of the s	
(Question #8:
1 1	include ryine 32.inc
	odata
	msg byte 011101016
	prompt1 byte "Message has even parity";)
	prompt 1 byte "Message has even parity!", 0 prompt 2 byte "Message hers odel parity!", 0
- Carrie Ly Carrier St	·code
	main prox
	mov eax, o
	mov al, msg add al, 000000106
	jp and not even
	inp even
	noteven:
	mov edx, offset prompt2
No. makip delicar oppositions and	call writestring
and the control distribution with purpose and the control distribution and	jmp sendd
The sale supplementary	even:
ari kan ⁵ adigating papang pang pang manggan	mov edn, offset prompt1
	coll writestring
hi dhan a sama dha a sa	
and the same of th	exit
	main endp
in the second	end main