4E Prelim Computing Paper 1 2018

1	A <u>computer processor</u> carries out the instructions in a <u>computer</u> <u>program</u> . The <u>memory</u> is where data and instructions are stored ready to be processed. The <u>address bus</u> identifies the location of data. The <u>data bus</u> transports the data from one location to another.	[5]
	Total	[5]
2	Volatile data storage: Processor register, RAM (any one)	[2]
	Non-volatile data storage: ROM, Any external storage (any one)	
	Total	[2]
3	Inputs: - 24 student names - 24 marks for Physics Outputs: - Name of the student with the lowest score - score for that student Processes required: - Store input data in arrays/lists - Search for the student with the lowest mark/sort the arrays in mark order	[6]
	Total	[6]
4	 Step 1: If <i>Number</i> is less than <i>Divisor</i>, proceed to Step 3. Otherwise, proceed to Step 2. Step 2: Subtract <i>Divisor</i> from <i>Number</i> and then proceed to Step 1. Step 3: Output <i>Number</i> as the final answer. 	[3]
	Total	[3]

5						[5]
	num	sum	n	rem	OUTPUT	
	371					
		0	371			
		1		1		
		1	37			
			01	7		
		344				
			3			
				3		
		371				
			0		трис	
					IRUE	
					Total	[5]
6	a constant: A	name with an a	associated val	ue that usually	/ cannot be	[1]
	changed while	e the program i	s running.			
	a variable. A name with an appropriated value that can be abar red while				[1]	
	a variable: A name with an associated value that can be changed while					[']
	an array: Sequ	lence where a	II the items h a	we the same	size and are	[1]
	arranged con	secutively in	memory.			
	arranged con	secutively in	memory.		Total	[3]
7	arranged con	secutively in	memory.		Total	[3]
7	<pre>arranged con a = int(ing b = int(ing</pre>	put ("Please	enter an	integer: "))	[3]
7	<pre>arranged con a = int(ing b = int(ing total = a -</pre>	put("Please put("Please put("Please	enter an another	integer: ") ther intege) er: "))	[3]
7	<pre>arranged con a = int(ing b = int(ing total = a - if a == b:</pre>	put("Please put("Please put("Please + b	e enter an a enter ano	integer: ") ther intege	Total) er: "))	[3]
7	<pre>arranged con a = int(ing b = int(ing total = a - if a == b: print()</pre>	put("Please put("Please put("Please + b "Twice the	<pre>memory. e enter an a enter ano sum:", 2*te</pre>	integer: ") ther intege otal)	Total) er: "))	[3]
7	<pre>arranged con a = int(ing b = int(ing total = a - if a == b: print(' else:</pre>	put("Please put("Please put("Please + b "Twice the	memory.	integer: ") ther intege otal)	Total) er: "))	[3]
7	<pre>arranged con a = int(inp b = int(inp total = a - if a == b: print(' else: print('</pre>	put("Please put("Please put("Please + b "Twice the "Sum:", tot	memory.	integer: ") ther intege otal)	Total	[3]
7	<pre>arranged con a = int(inp b = int(inp total = a - if a == b: print(' else: print('</pre>	put("Please put("Please t b "Twice the "Sum:", tot	memory.	integer: ") ther intege otal)	Total	[3]
7	<pre>arranged con a = int(inp b = int(inp total = a - if a == b: print(' else: print('</pre>	secutively in put("Please put("Please + b "Twice the "Sum:", tot	memory.	integer: ") ther intege otal)	Total	[3]
7	<pre>arranged con a = int(ing b = int(ing total = a - if a == b: print(' else: print('</pre>	secutively in put("Please put("Please + b "Twice the "Sum:", tot	memory.	integer: ") ther intege otal)	Total	[3]
7	<pre>arranged con a = int(ing b = int(ing total = a - if a == b: print(' else: print(' length check:</pre>	put("Please put("Please t b "Twice the "Sum:", tot	memory.	integer: ") ther intege otal) ut data is not t	Total	[3] [9] [9] [1]
7	<pre>arranged con a = int(ing b = int(ing total = a - if a == b: print(' else: print(' length check: short.</pre>	put("Please put("Please but("Please "Twice the "Sum:", tot	memory.	integer: ") ther intege otal) ut data is not t	Total) er: ")) Total oo long or too	[3] [9] [9] [1]
7	<pre>arranged con a = int(ing b = int(ing total = a - if a == b: print(' else: print(' length check: short.</pre>	secutively in put("Please put("Please + b "Twice the "Sum:", tot	memory.	integer: ") ther intege otal) ut data is not t	Total) er: ")) Total oo long or too	[3] [9] [9] [1]
7	<pre>arranged con a = int(ing b = int(ing total = a - if a == b: print(' else: print(' length check: short. range check: F</pre>	<pre>put("Please put("Please put("Please + b "Twice the "Sum:", tot Process of ens Process of ens</pre>	memory.	integer: ") ther intege otal) ut data is not t	Total) er: ")) Total oo long or too n the required	[3] [9] [9] [1] [1]
7	<pre>arranged con a = int(ing b = int(ing total = a - if a == b: print(' else: print(' length check: short. range check: H range of value</pre>	Process of enses	memory.	integer: ") ther intege otal) ut data is not t	Total) er: ")) Total oo long or too n the required	[3] [9] [9] [1] [1]
7	<pre>arranged con a = int(ing b = int(ing total = a - if a == b: print(' else: print(' length check: short. range check: H range of value</pre>	put("Please put("Please out("Please + b "Twice the "Sum:", tot Process of ens S.	memory.	integer: ") ther intege otal) ut data is not t	Total) er: ")) Total oo long or too h the required	[3] [9] [9] [1] [1]

9	Error 1 : Total = 500	[1]
	Correction: Total = 0	[1]
	Error 2: WHILE Counter > 11 DO	[1]
	Correction: WHILE Counter < 11 DO	[1]
	Error 3 : Total = Total - Number[Counter]	[1]
	Correction: Total = Total + Number[Counter]	[1]
	Error 4: Average = Total / Counter	[1]
	Correction: Average = Total / 10	[1]
	Total	[8]
10		[5]
	malicious code installed on the hard drive of a user's computer or on the web server; this code will re-direct user to a fake web site without their consent	
	software that gathers information by monitoring key presses on a user's computer and relays the information back to the person who sent the software	
	program or code that replicates itself and is designed to amend/delete/copy data and files on a user's computer without their consent	
	the act of gaining illegal access to a computer system without the owner's consent	
	creator of code sends out a legitimate-looking email in the hope of gathering personal and financial data from the recipient; it requires the email or attachment to be opened first	
	Total	[5]
44	P = 11 = 1011	[0]
11 a	$B_{16} = 11_{10} = 1011_2$	
	$C_{16} = 12_{10} = 1100_2$	
	Binary number = 1011 1100	

11 b	$2\overline{)401} = 200R1$		
	2)200 = 100R0		
	2)100 = 50R0		
	$2\overline{)50} = 25R0$		
	2)25 = 12R1		
	$2\overline{)12} = 6R0$		
	$2\overline{)6} = 3R0$		
	$2\overline{)3} = 1R1$		
	2)1 = 0R1		
	Binary number = 110010001		
11	$16\overline{)11001} = 687R9$		
	$16)\overline{687} = 42RF$		
	$16\overline{)42} = 2RA$		
	$16\overline{)2} = 0R2$		
	Hexadecimal number = 2AF9		
		Total	[6]
12			[6]
	PNOT		
	T NOT		
		Total	[6]

13	=B5/C5			[1]		
a						
13	3 = (D2 + D3 + D4 + D5 + D6)/5 OR				[1]	
b	= AVERAGE(D2:D6) OR = SUM(D2:D6)/5					
10						
13			E	F	G	[4]
С			Percent	Discount	Discounted price	
	1	1	discount	amount	per bottle (\$)	
		<u> </u>	<u>(%)</u>	(\$) = D0 * E0/400	- 50 50	-
	1	2	10	= B2 * E2/100	= B2 - F2	-
		3 1	15	- B4 * E4/100	- B3 - F3	-
		4 5	10	= B5 * E5/100	= B4 - F4 = B5 - F5	-
		6	5	= B6 * E6/100	= B6 - F6	-
		•	0	- 00 20/100	- 80 - 10	1
					Tota	al [6]
14	A network allows a group of computers to make use of shared				[2]	
а	res	sources	such as printe	ers or files.		
	Depending on the network's configuration, every user who logs on to the network may have access to the Internet.					е
	Software can be stored on the central server of a network and deployed to other computers over a network.					1
	Data files can be stored on a central server for ease of access and backup purposes.					
	Computers in the same network are often able to share instant					
	messages and emails for communication.					
14	- C	counts t	he number of c	ones in each byte of da	ta	[4]
b	- parity bit is set so there are an even/odd number of ones before the data is transmitted.					
	- If the number of ones is not even/odd when the data has been transmitted					
	- then an error has occurred					
					Tota	al [6]



15	Pseudo-code	[8]
15	<pre>Pseudo-code number = [] negative = 0 positive = 0 index = 0 WHILE index < 20 INPUT number[index] IF number[index] < 50 OUTPUT index ENDIF IF number[index] < 0 negative = negative + 1 ELSE positive = positive + 1 ENDIF index = index + 1 ENDWHILE OUTPUT "Number of negative numbers: " + negative]</pre>	[8]
	OUTPUT "Number of positive numbers: " + positive }	
	Total	[8]