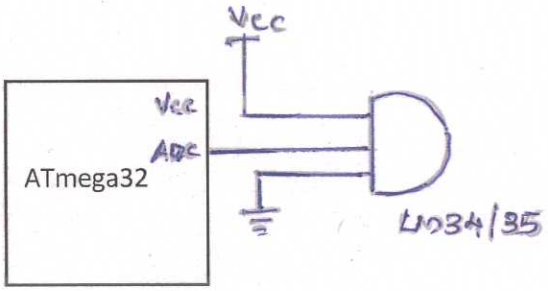
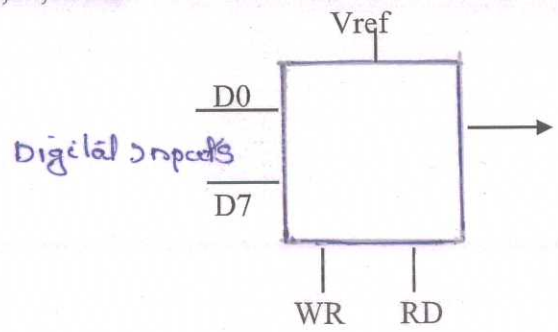
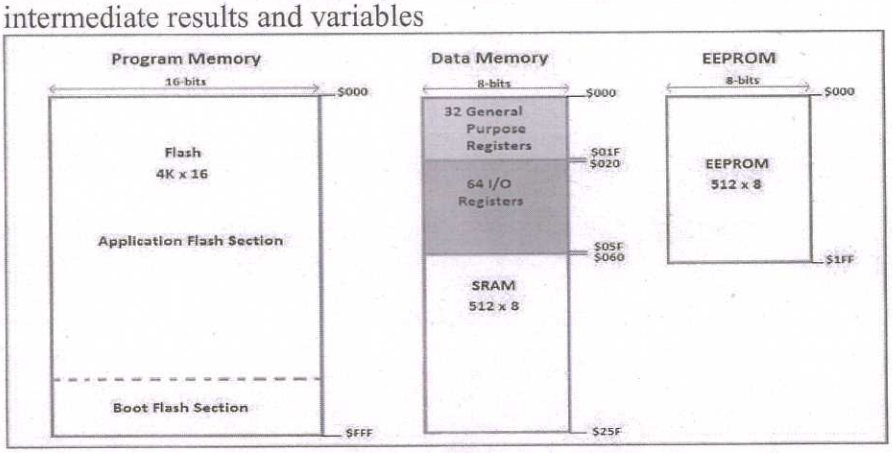
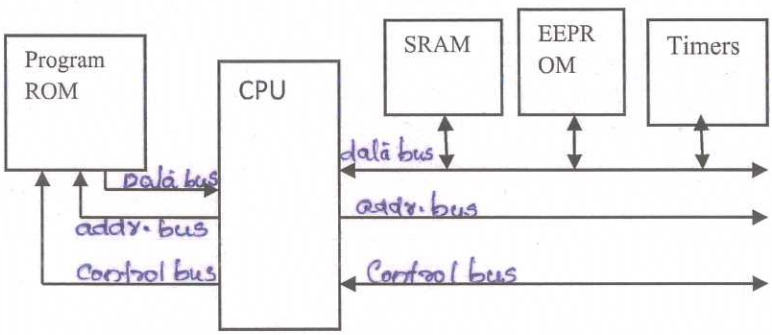
SCHEME OF EVALUATION

Revision:2015 SIXTH SEMESTER DIPLOMA EXAMINATION IN ENGINEERING/TECHNOLOGY
Course Titles: MICROCONTROLLERS (6132)

Qst No.	Scoring Indicator	Split up score	Sub Total	Total
I	PART A			
1.	Hex, Binary, Decimal, ASCII	1/2x4	2	
2.	AVR data memory consists of SRAM which is used for temporarily storing and keeping intermediate results and variables	2	2	
3.	*This instruction tests a single bit in an I/O register and skips the next instruction if the bit is set. *Ex: SBIS R17, 3 - It checks the third bit of R16 register value. If it is set then skip the next instruction	1 1	2	10
4.	*AVR Timers can be used as timers and counters. *It can be used for creating delays.	1 1	2	
5.	*RS232 is a serial communication interfacing standard. *It allows compatibility among data communication equipment made by various manufacturers.	1 1	2	
II	PART B			
1.	*Microcontroller is a small computer on a single IC. *It contains a processor core, ROM, RAM and I/O pins dedicated to perform various tasks, * it does not need any external circuits to do its task *so microcontrollers are heavily used in embedded systems.	3x1		
	*Microprocessor has only a CPU inside them in one or few Integrated Circuits. *Like microcontrollers it does not have RAM, ROM and other peripherals. * They are dependent on external circuits of peripherals to work, * But microprocessors are not made for specific task but they are required where tasks are complex and tricky like development of software's. (any 3differences)	3x1	6	6
2.	• It is 8 bit , • RISC, • single chip, • Harvard architecture, • On chip program ROM, data RAM, data EEPROM, Timers and I/O ports. • It has additional features like ADC, PWM and different kinds of serial interface. (Any 6 points)	6x1	6	6

3.	Char, unsigned char, int, unsigned int, long , unsigned long, float, double (list any three – 3*1/2 Example - 3x1/2 Explanation -1+1+1)	3x 1/2 3x 1/2 1+1+1	6	6
4.	*Using simple for loop: crystal frequency connected to the XTAL1, XTAL2 input pin is the most important factor in time delay calculation., Second factor is the compiler used to compile the C program. Ex: For (i=0; i<500; i++); *Using AVR timer- load TCNTx, set corresponding bits of TCCRx *Using built in delay functions Functions: _delay _ms( ) and _delay_us( ) *defined in delay.h	3x2	6	6
5.	*Timer can be used as a counter by giving an external pulse to the timer * increment TCNTx register by using this external pulse. *This can be done by using T0 pin in Timer0 and T1 in Timer1. *In Atmega 32 PB.0 is the alternative function of T0. *For selecting these pins set the bits of CS00, CS01 and CS02 of TCCR0 as 6 or 7. *similarly set the value of CS10,CS11,CS12 as 6 or 7 in case of Timer1	6*1	6	6
6.	*LM35 is a precision integrated circuit temperature sensor whose output voltage is linearly proportional to the Celsius temperature, *It does not require any external calibration, *It is a 3-terminal device that provides analog voltage proportional to the temperature. Higher the temperature, higher is the output voltage, *LM34 is a precision integrated circuit temperature sensor whose output voltage is linearly proportional to the Fahrenheit temperature.  	3*1 Fig:3	6	6
7.	Digital to analog converter coverts digital pulses to analog signals. Resolution is an important criterion which is a function of binary inputs.	3		

	Number of analog output levels is equal to 2^n, Common resolution are 8,10,12 bits.  (or any related figure)	3	6	6
IIIa.	*AVR microcontroller's memory is divided into Program Memory and Data Memory. *Program Memory (ROM) is used for permanent saving program being executed, * Data Memory (RAM) is used for temporarily storing and keeping intermediate results and variables. *Data memory consists of :General purpose registers, I/O Memory, Extended I/O ,Internal SRAM *There are 32 general purpose working 8-bit registers (R0-R31). These registers have the shortest (fastest) access time, which allows single-cycle Arithmetic Logic Unit (ALU) operation. *I/O Memory space contains addresses for CPU peripheral function, such as Control registers, SPI, and other I/O functions. *Due to the complexity; some AVR microcontrollers with more peripherals have Extended I/O memory, which occupies part of the internal SRAM. *Extended I/O memory is MCU dependent. Storing data in I/O and Extended I/O memory is handled by the compiler only. Users can not use this memory space for storing their data. Internal SRAM (Data Memory) is used for temporarily storing and keeping intermediate results and variables  (for drawing data memory- give full(2) marks)	List-2 Explanation -5 Fig: 2	9	9
III b.	*Branch instruction transfers control to different location. *There are conditional and unconditional branches *JMP and RJMP are unconditional jumps. It transfers control without any	1 1 2		

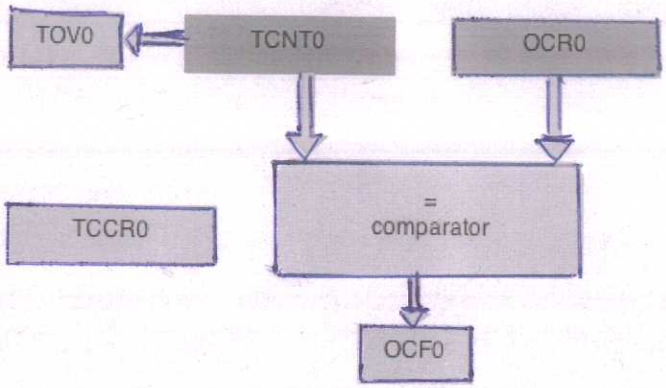
	condition. *In Conditional jump, it transfers control based on some condition BRLO – branch if C=1 BRSH-Branch if C=0 BREQ- Branch if Z=1 BRNE- Branch if Z=0	1 1	6	6
IV a.	*The Harvard architecture offers separate storage and signal buses for instructions and data. This architecture has data storage entirely contained within the CPU, and there is no access to the instruction storage as data. *Computers have separate memory areas for program instructions and data using internal data buses, allowing simultaneous access to both instructions and data. *Programs needed to be loaded by an operator; the processor could not boot itself. In Harvard architecture, there is no need to make the two memories share properties.  RISC- *fewer instructions with simple constructs, *they can be executed much faster within the CPU without having to use memory, *each instruction is to be executed by hardware, * few addressing modes, * fixed instruction format, * large number of registers. (any 4 points)	2 2 4	8	8
IV b.	*Call instructions leave a return address on to the stack * jump to the call destination. *There are variants of call (call subroutine), rcall (relative call subroutine), icall (indirect call to Z) and eicall (extended indirect call to Z). *The stack pointer (SP) needs to be set up prior to any subroutine calls. *The stack can be used by call instructions to store the return address, which is the address following the call instruction. *After RET instruction in subroutine it pops the stack content to PC . * return back to the calling function.	3 4	4+3	7
V a.	*Atmega 32 has 4 I/O ports each 8 bits. * PORTA, PORTB, PORTC, PORTD * Each Port is associated with 3 registers for configuration(Input/Output),	Explanation -5		

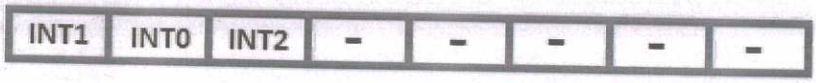
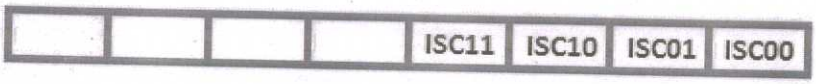
	read and write operation. DDRx: Data Direction Register, Used to configure the respective PORT as output/input DDRx=0Xff it makes port as output DDRx=0X00 it makes port as input Ex: in c DDRA=0x00; PORTA as input PORTx: It is used to write the data to the Port pins Ex: PORTA=0X55 PINx: Used to Read the data from the port pins Ex: x=PINA; Read PORTA and stores in x	Eg:1+1+1	8	8

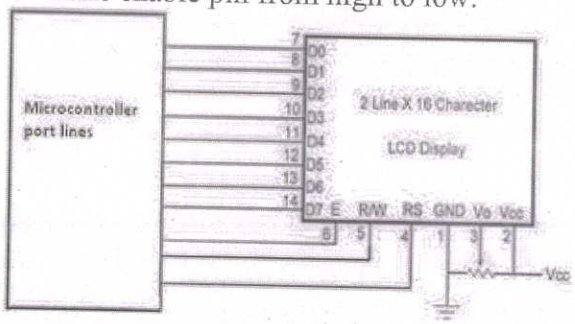
Vb.	#include <avr/io.h>
	int main(void)
	{
	unsigned char k;
	un signed char x='3';
	un signed char y='6';
	DDRC= 0xFF;
	x = x & 0x0F;
	x = x<<4;
	y= y & 0x0F;
	k = x | y;
	PORTC=k;
	return 0;
	} correct logic-3, use of logical operators-2, header file-1, syntax-1
		3+2+1+1	7	7

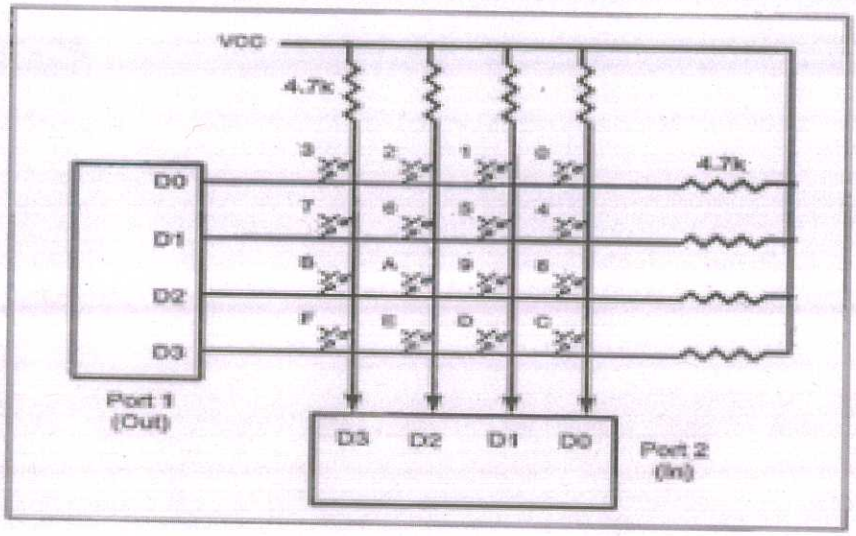
VIa.	#include <avr/io.h>
	int main(void)
	{
	unsigned char x;
	DDRA= 0x00;
	DDRB=0xFF;
	x = PINA;
	PORTB=x;
	return 0;}
	Setting ports –2, reading&writing-2, header file-1, syntax-2,logic -1
		2+2+1+2+1	8	8

VIb.	#include <avr/io.h>
	int main(void)
	{
	unsigned char x;
	DDRD= 0xFF;
	While(1)
	{
	PORTD = PORTD | 0b00001000 ;
	PORTD = PORTD & 0b11110111;
	}
	return 0;}
	Setting ports –1, writing -2, header file-1, syntax-1, logic -2
		1+2+1+1+2	7	7

VII a.	*TIMER0 is a 8 bit timer of AVR *It has registers such as TCNT0, TCCR0, OCR0 *And flag such as TOV0,OCF0 *TCNT0: it is 8 bit register; The value of the counter is stored here and increases/decreases automatically. Data can be both read/written from this register. *TCCR0- Timer counter control register for setting modes and timer clock selector. *OCR0- content of this register is compared with the content of TCNT0. When match ours OCF0 flag will be set. *When Timer overflows, its TOV0 flag will be set.  	1 1 1 1 1 1 1 Fig:3	10	10
VII b.	*If two interrupts are activated at the same time, interrupt with higher priority is served first. *The priority of each interrupt is related to the address of that interrupt in the interrupt vector. *Interrupt with lower address has higher priority. *When AVR begins to execute an ISR, it disables the I bit of the SREG register, *It causes all the interrupts to be disabled	5	5	5
VIII a.	1.Load TCNT0 register with initial count value 2.Load the value into TCCR0 register 3. Keep monitoring timer overflow flag to see if it is raised. Get out of the loop when it is high 4.Stop the timer TCCR0= 0x00; 5. clear the TOV0 flag by TIFR=0x01; 6. Go back to step 1	5*1	5	
VIII b.	*External hardware interrupts are External Interrupt 0 (**INT0**), External Interrupt 1 (**INT1**) and External Interrupt 2 (**INT2**) *GICR: It helps to enable the External interrupts	List-3 1		

	 *INT0 and INT1 are level and edge triggered. This can be configured using MCUCR. MCUCR  *For INT0 selection of trigger event will be based on ISC01 and ISC00 ISC01 ISC00 0 0 Low level 0 1 Any logical change 1 0 falling edge 1 1 rising edge *For INT1, it is based on ISC10 and ISC11 ISC11 ISC10 0 0 Low level 0 1 Any logical change 1 0 falling edge 1 1 rising edge *INT2 is Edge triggered. ISC2 of MCUCSR bit is use to configure event triggers of INT2. 1- Rising edge, 0-Falling edge	2 1 1 2	10	10
IX a.	*LCD stands for Liquid Crystal Display, is an electronic device which is used for data display. LCDs are preferable over seven segments and LEDs as they can easily represent data in form of alphabets, characters, numbers or animations. *In 8-bit mode we send command to LCD by using eight data lines (D0-D7) while in 4-bit mode we use four data lines (D5-D7) for sending command and data. These data lines can be connected to any port of Atmega32. *Basically there are two registers, command and data. When we are giving command to LCD, we select command register and when we are sending data to LCD for display, we select data register. Command is an instruction given to LCD in order to perform required function according to the given command. In order to display textual information, data is send to LCD.	1 2		

	SENDING COMMANDS ON LCD: For sending commands on LCD we have to write command on data pins. For this, selects: RS = 0 - selects command register, RW = 0 -selects write operation, E - make enable pin from high to low ***SENDING DATA ON LCD:*** For sending data on LCD we have to write data on data pins. For this, selects:RS = 1 - selects data register, RW = 0 -selects write operation E -make enable pin from high to low.  (Any related figure)	1 1/2 1 1/2 Fig:4	10	10
IX b.	PIN Description: 1 VCC — Supply pin (+5V DC) 2 VDD — Ground pin 3 VEE — Contrast pin 4 RS — Register selection pin (either data or command)RS=0: Command Register , RS=1: Data Register 5 RW — Selects Read or Write operation RW=0: for write RW=1: for read 6 E — Enable pin 7 D0-D7 — Data pins	Pin Description-5		5
X a.	*The keypad here has four columns and four rows, *for identification of button pressed, we are going to use cross reference method. * first, either connect all columns or all rows to vcc , *if rows are connected to common vcc, we are going to take the columns as inputs to controller. *Now if button one is pressed, a current flows through the circuit. So we have C1 high, for a button press. *At this very moment, shift the power and input ports that is, power the columns and take rows as inputs	Explanation -6		

	 (or Any related figures)	Fig:4	6+4	10
X b.	*Analog to digital converter, *Atmega32 ADC is 10 bit, *It has 8 analog input channels, *The converted output binary data is held by two special function registers called ADCL and ADCH.6 bits of ADCH and ADCL registers are unused. * Vref can be connected to AVCC, internal 2.56V reference, or external AREF pin. (any 5 points)	5	5	5