**ANSWER 1**

A loop becomes an infinite loop if a condition never becomes false. The for loop is used for this purpose. Since none of the three expressions that form the 'for' loop are required, we can make an endless loop by leaving the conditional expression empty.   
like this :-  
  
**FOR**

#include <stdio.h>  
Int main()  
{

For ( ; ; )

{

Printf(“this is example of infinite loop using for loop”)

}

Return 0;

}

**WHILE**

#include <stdio.h>

int main()

{

int i = 0;

while(1)

{

i++;

printf("i is : this is infinite loop in while ",i); // a null statement

}

return 0;

}

**DO WHILE**

#include <stdio.h>

int main()

{

int i=0;

do

{

i++;

printf("i is : this is infinite loop in do while ",i);

}while (1);

}

**ANSWER 8**

**SWITCH STATEMENT**  
The switch statement allows us to execute one code block among many alternatives values.   
Each value is called a case, and the variable being switched on is checked for each switch case.  
EXAMPLE  
here I have given a example of calculator using switch   
#include <stdio.h>

int main() {

char operator;

double n1, n2;

printf("Enter an operator (+, -, \*, /): ");

scanf("%c", &operator);

printf("Enter two operands: ");

scanf("%lf %lf",&n1, &n2);

switch(operator)

{

case '+':

printf("%.1lf + %.1lf = %.1lf",n1, n2, n1+n2);

break;

case '-':

printf("%.1lf - %.1lf = %.1lf",n1, n2, n1-n2);

break;

case '\*':

printf("%.1lf \* %.1lf = %.1lf",n1, n2, n1\*n2);

break;

case '/':

printf("%.1lf / %.1lf = %.1lf",n1, n2, n1/n2);

break;

default:

printf("Error! operator is not correct");

}

return 0;

}  
  
**ANSWER 7**  
nested if else nothing but using else operator again and again   
example :-

#include <stdio.h>

int main()

{

int var1, var2;

printf("Input the value of var1:");

scanf("%d", &var1);

printf("Input the value of var2:");

scanf("%d",&var2);

if (var1 != var2)

{

printf("var1 is not equal to var2\n");

//Nested if else

if (var1 > var2)

{

printf("var1 is greater than var2\n");

}

else

{

printf("var2 is greater than var1\n");

}

}

else

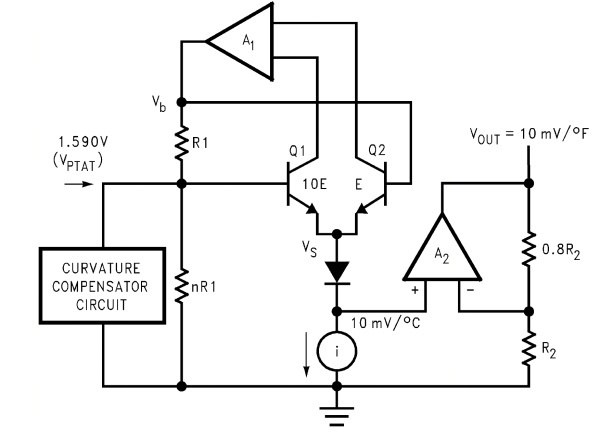
{

printf("var1 is equal to var2\n");

}

return 0;

}  
  
**ANSWER 9**  
  
The LM35 is an integrated circuit sensor that can be used to measure temperature with an electrical output proportional to the temperature (in °C). It can measure temperature more accurately than a using a thermistor. The sensor circuitry is sealed and not subject to oxidation. The LM35 generates a higher output voltage than thermocouples and may not require that the output voltage be amplified. The LM35 has an output voltage that is proportional to the Celsius temperature. The scale factor Is 0.01V/°C  
If we dig a bit deeper,  
The working can be described as follows:  
There are two transistors in the centre of the drawing. One has ten times the emitter area of the other. This means it has one tenth of the current density, since the same current is going through both transistors. This causes a voltage across the resistor R1 that is proportional to the absolute temperature, and is almost linear across the range we care about.   
The amplifier at the right converts absolute temperature (measured in Kelvin) into either Fahrenheit or Celsius, depending on the part (LM34 or LM35). The little circle with the "i" in it is a constant current source circuit. The two resistors are calibrated in the factory to produce a highly accurate temperature sensor. The integrated circuit has many transistors in it -- two in the middle, some in each amplifier, some in the constant current source, and some in the curvature compensation circuit. All of that is fit into the tiny package with three leads.



**ANSWER 10**

LDR interface on dark surroundings  
light gets turned on when its dark environment and gets turned off when its bright environment

int sensorPin = 0;

int ledPin = 13;

int sensordark=0;

int sensorbright = 600;

void setup()   
{

pinMode(13, OUTPUT);

}

void loop()  
 {

sensordark = analogRead(sensorPin);

delay(100);

if (sensordark<sensorbright)

{

digitalWrite(13, HIGH);

}  
 else

{

digitalWrite(13, LOW);

}

}

**ANSWER 2**  
void setup()

{

pinMode(13, OUTPUT);

int i=0;

while(i<6 )

{

i++;

digitalWrite(13, LOW );

delay(1000);

digitalWrite(13, HIGH);

delay(1000);

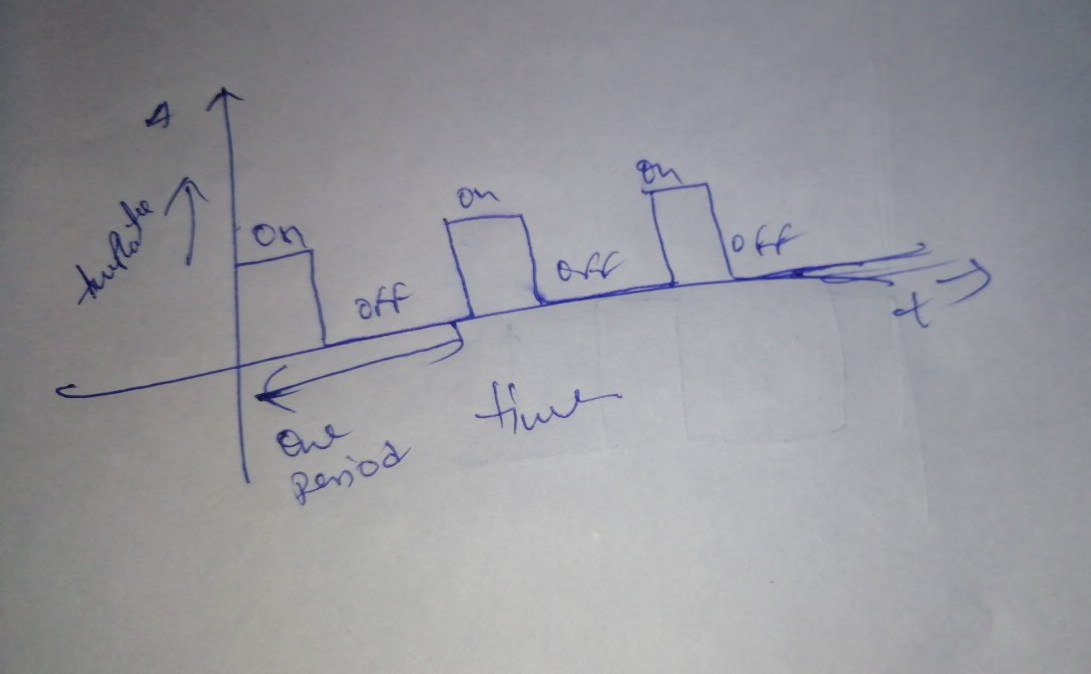
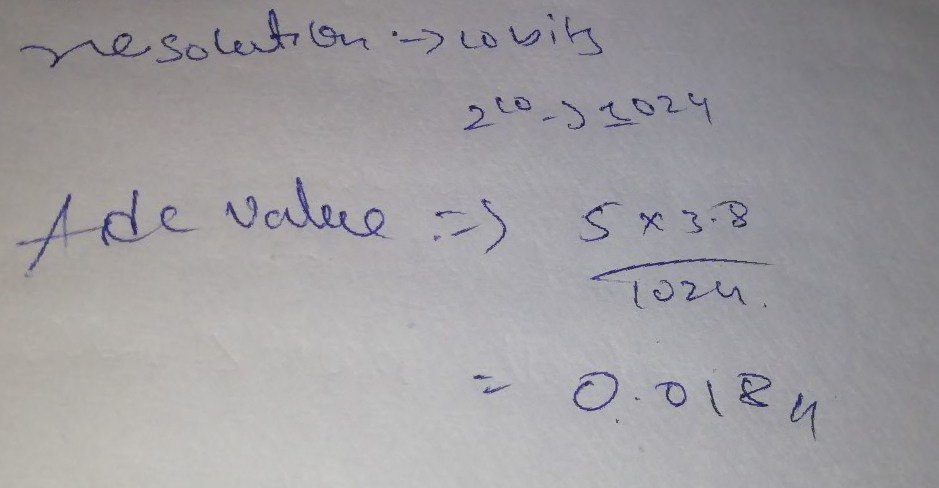
}

}

void loop()

{

}

**ANSWER 4**  
  
  
  
**ANSWER 3**  
  
power supply = 5V  
input voltage = 3.8V  
resolution = 10 bits  
  
  
  
  
  
  
  
**ANSWER 5**  
  
#define ledPin D1

#define pushbutton D2

ICACHE\_RAM\_ATTR void detect(){

Serial.println("Interrupt Detected");

}

void setup() {

pinMode(ledPin,OUTPUT);

attachInterrupt(digitalPinToInterrupt(pushbutton),detect, RISING);

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

int i=0;

while (1){

i++;

digitalWrite(ledPin,HIGH);

delay(1000);

digitalWrite(ledPin,LOW);

delay(1000);

}

}  
 **ANSWER 6**  
  
  
#define ledPin D1

#define pushbutton D2

#define pir D3

ICACHE\_RAM\_ATTR void detect()  
{

Serial.println("Interrupt Detected");

}

void setup() {

pinMode(ledPin,OUTPUT);

attachInterrupt(digitalPinToInterrupt(pushbutton),detect, RISING);

attachInterrupt(digitalPinToInterrupt(pir),detect, RISING);

Serial.begin(9600);

}

void loop() {

int i=0;

while (1)  
{

i++;

digitalWrite(ledPin,HIGH);

delay(1000);

digitalWrite(ledPin,LOW);

delay(1000);

}

}