1. Express the following statements in C language
   1. If a is greater than 5.0, set a as 5.0
   2. If a or b equals 10, c=100

2. Write the output of the following C program
   ```c
   void main()
   {
       int i, j, m, n;
       i=8; j=10;
       m=++i;
       n=j++;
       printf("%d,%d,%d,%d", i, j, m, n);
   }
   ```

3. Correct the following C statements
   1. `scanf(%d,&a);`
   2. `if(0<a<10) b=100;`
   3. `for(i=1,i<10,i=i+5) a++;`
C Programming

FUNCTIONS
They are blocks of code that chop up a long program into named sections.

You can either use the built-in library function or create your own functions (user defined functions).

Functions accept parameters (no limit) and return a result.

Ordering is not needed when functions are arranged in a program.
In C program, a special function called main() is the primary function that calls all other functions.
C Programming

#include <stdio.h>
void write()
{
    printf("Hello world !\n");
}
int main()
{
    write();
    return 0;
}

Note: There is no “return” statement for “void”
```c
#include <stdio.h>

float cube(float x)
{
    return x*x*x;
}

int main()
{
    float x, x3;
    printf("Enter x = "); scanf("%f", &x);
    x3 = cube(x);
    printf("The cube of x is %f \n", x3);
    return 0;
}
```

Note: The function type has been declared (if “void” not used)
Another Example

\[ z \equiv x^y \]
then by taking the natural logarithm of both sides, one obtains

\[ \log z = \log x^y = y \log x \]
so that
\[ z = \exp(y \log x) \]

We will program this as a special function
```c
#include <stdio.h>
#include <math.h>

float power(float x, float y)
{
    return exp(y*log(x));
}

int main()
{
    float x,y; printf("Enter x and y separated by space =");
    scanf("%f %f",&x,&y);
    if (x<0)
    {
        printf("x must be positive !!\n");
        return 0;
    }
    printf("%f to power of exponent %f is %f.\n", x, y, power(x,y));
    return 0;
}

$ gcc -lm example-5-00.c
```
```c
#include <stdio.h>

int add(int i, int j)
{
    int k;
    k = i+j;
    return k;
}

int main()
{
    printf("%d\n", add(3, 4));
    return 0;
}

$ gcc example-5-01.c
```
C Programming

- C lets you place function prototypes at the beginning of (actually, anywhere in) a program.

- Try compiling this program …..

```c
#include <stdio.h>
int add (int, int); /* function prototype for add */
int main()
{
    printf("%d\n", add(3, 10));
    return 0;
}
int add(int i, int j)
{
    return i+j;
}
```

Note: How the function prototype is declared

$ gcc example-5-02.c$
Variables used inside the function block are all local. They do not carry values outside the function.

A function can be recursive. It can call itself.
C Programming

Recursive
/ Compute b = 1 - 1/2 + 1/3 - 1/4 + ... */

#include <stdio.h>
#include <math.h>
float invn(int);
int main()
{
    int n; float b;
    printf("Enter the number of terms in the series \n") scanf("%d",&n);
    if(n>0)   {
        printf("The value of b is %f (Recursive) \n", invn( n ));
    }
    return 0;
}

float invn( int n)
{
    if (n <=0 ) return 0,0;
    return -pow(-1,n)/n + invn(n-1);
}

$ gcc example-5-03.c
C Programming

Cast Operators

Any operation between values of different types is performed by converting the lower type to the higher type

```
int < float < double
```

If a(float) & b(integer) then for a*b, ‘b’ becomes float automatically
#include <stdio.h>
int main()
{
    int a=3; float b=5.0;
    int c=5; float d;
    d = a/b;
    printf("%f \n", d);

    c = a/b;
    printf("%d \n", c);
    printf("%.1f \n", (float) a/b);
    return 0;
}
C Programming

Constant
You can define a **symbolic constant** by using preprocessor

It is customary to use upper case

**Example: Find the area of a circle by using pi as a constant**

```c
#include <stdio.h>
#define PI 3.141592        /* define pi */
int main()
{
    float r;
    printf("Enter radius = ");
    scanf("%f", &r);
    printf("The area of circle is=%f \n", r*r*PI);
    return 0;
}
```

$ gcc example-5-05.c$