University of Mumbai

Program: _First Year (All Branches) Engineering - SEM-II
Curriculum Scheme: Rev 2019
C-Programming

Question Bank

==========

==========

Q1.	**Choose the correct option for following questions. All the Questions are compulsory and carry equal marks**
1.	Which storage class is called as default storage class ?
Option A:	**auto**
Option B:	register
Option C:	static
Option D:	extern
2.	What inbuilt function should be used to return a value rounded up to the next higher integer ?
Option A:	floor
Option B:	malloc
Option C:	puts
Option D:	**ceil**
3.	In the following initialization what is value of A[5] ? int A[10] = {9, 8, 7, 6, 5, 4, 3, 2, 1, 0};
Option A:	5
Option B:	**4**
Option C:	3
Option D:	2

4.	What is the output for the following code ?
	int main()
	{
	int a=5,i;
	i!=a >10;
	printf("i=%d",i);
	return 0;
	}

Option A:	**i=0**
Option B:	i=10
Option C:	i=110
Option D:	i=1

5.	How many times will the following while-loop repeat, i.e., how many x are printed?
	int main()
	{
	int i = 5;
	while(i> 0)
	{
	printf("x");
	i--;
	}
	return 0;
	}

Option A:	2
Option B:	3
Option C:	4
Option D:	**5**
6.	Which among the following is an exit controlled loop ?
Option A:	for
Option B:	while
Option C:	**do… while**
Option D:	if…else

7	What is another name for 1-D arrays ?
Option A:	**Linear arrays**
Option B:	Lists
Option C:	Horizontal array
Option D:	Vertical array
8	Which of the following operators takes only integer operands?
Option A:	+
Option B:	*
Option C:	/
Option D:	**%**
9	What is value of a in following expression? int a = 10 + 4.867;
Option A:	a=10
Option B:	a=14.867
Option C:	**a=14**
Option D:	a=4
10	C programs are converted into machine language with the help of ----------.
Option A:	an editor
Option B:	an Assembler
Option C:	**a compiler**
Option D:	an operating system

11	What is the output of the program.?
	int main()
	{
	float a = 45;
	printf("%f", a);
	return 0;
	}

Option A:	45
Option B:	45.0
Option C:	**45.000000**
Option D:	0.000000
12	Which among the following is a Conditional Operator in C ?
Option A:	**?:**
Option B:	:?
Option C:	<=

Option D:	>=

13	What is the output of the C statement.?
	int main()
	{
	int a=0;
	a = 5<2 ? 4 : 3;
	printf("%d",a);
	return 0;
	}

Option A:	4
Option B:	**3**
Option C:	5
Option D:	2
14	Recursion is a process in which a function calls _____.
Option A:	**itself**
Option B:	another function
Option C:	main() function
Option D:	sub program
15	What is the Format specifier used to print a character in C.?
Option A:	%s
Option B:	**%c**
Option C:	%C
Option D:	%w
16	Which of the following is not a relational operator?
Option A:	>=
Option B:	>>
Option C:	==
Option D:	!=
17	Which one of the following is a valid C expression?
Option A:	**int my_number=1000;**
Option B:	int my-number=1000;
Option C:	int my@number=1000;
Option D:	int @mynumber=1000;

18	What will be the output of the following C code?
	#include <stdio.h>
	int main()
	{
	int a = 1, b = 1, c;
	c = a++ + b;
	printf("a=%d, b=%d", a, b);
	}

Option A:	a=1, b=1
Option B:	**a=2, b=1**
Option C:	a=2, b=2
Option D:	a=1, b=2

19	What will be the output of the following C code?
	
	#include <stdio.h>
	  void main()
	  {
	  int x = 5;
	  if (x == 5)
	  printf("hi\n");
	  else
	  printf("how are u\n");
	  printf("hello\n");
	  }

Option A:	hi
Option B:	**hi** **hello**
Option C:	how are you hello
Option D:	how are you

20	What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input).
	#include <stdio.h>
	  void main()
	  {
	  int ch;
	  printf("enter a value between 1 to 2:");
	  scanf("%d", &ch);
	  switch (ch)
	  {
	  case 1:
	  printf("1\n");
	  break;
	  printf("hi");
	  default:
	  printf("2\n");
	  }
	  }

Option A:	**1**
Option B:	1 hi
Option C:	hi
Option D:	2

21	What will be the output of the following C code?
	
	#include <stdio.h>
	int main()
	  {
	  int i = 0;
	  while (i = 0)
	  printf("True\n");
	  printf("False\n");
	  }

Option A:	True
Option B:	**False**
Option C:	True

	False
Option D:	True (Infinite Times)

22	What will be the output of the following C code?
	
	#include <stdio.h>
	int main()
	{
	int x = 0;
	if (x = = 1)
	if (x = = 0)
	printf("inside if\n");
	else
	printf("inside else if\n");
	else
	printf("inside else\n");
	}

Option A:	inside if inside else
Option B:	inside else if
Option C:	inside if
Option D:	**inside else**
23	The value obtained in the function is given back to the main program by using which keyword?
Option A:	new
Option B:	**return**
Option C:	volatile
Option D:	static

24	What will be the output of the following C code?
	#include <stdio.h>
	void main()
	{
	m();
	m();
	}
	void m()
	{
	static int x = 5;
	x++;
	printf("%d", x);
	}

Option A:	5 5
Option B:	5 6
Option C:	6 6
Option D:	**6 7**
25	An array Index starts with.?
Option A:	**0**
Option B:	1
Option C:	-1
Option D:	2
26	What will be the output of the following C code? #include <stdio.h>

	void main()
	{
	char string[]={'E','X','A','M','\0'};
	printf("%s",string);
	}

Option A:	E
Option B:	EXAM0
Option C:	EXAM\0
Option D:	**EXAM**
27	Which one of the following is NOT an identifier?
Option A:	_cprogram
Option B:	c_program
Option C:	**20cprogram**
Option D:	cprogram20

28	What will be the output of the following program?
	int main()
	{
	int i=9;
	while(i++<10)
	printf("%d\n",i);
	return 0;
	}

Option A:	9
Option B:	**10**
Option C:	1
Option D:	11

29	What will be the output of the following program?
	int main()
	{
	int a,b,c,d,e,f,g,h,k;
	a=8, b=4, c=2, d=1, e=5, f=20;
	printf("%d\n",a+b-(c+d)*3%e+f/9);
	return 0;
	}

Option A:	**10**
Option B:	9
Option C:	8
Option D:	20

30	If a is a variable initialized to 1, how many times will the following loop be executed?
	while((a>0)&&(a<25))
	{
	loopbody
	a++;
	}

Option A:	25
Option B:	**24**
Option C:	20
Option D:	26

31	In an array a[2] [2] = {10,20,30,40,50,60}, then a[0] [1] is which element?
Option A:	10
Option B:	**20**
Option C:	30
Option D:	40

32	What will be the output of the following program?
	int main()
	{
	int a = 500, b = 100, c;
	if(!a >= 400)
	b = 300;
	else
	b=b+++b*a/b;
	c = 10;
	c=b<<1;
	c=c>>b+1;
	printf("b = %d c = %d\n", b, c);
	return 0;
	}

Option A:	B=600, c=3
Option B:	B=600, c=2
Option C:	B=600, c=1
Option D:	**B=600, c=0**
33	Which bitwise operator is used for turning off a particular bit in a number?
Option A:	|
Option B:	^
Option C:	&
Option D:	~

34	What will be the output of the following program?
	int i;
	int goodday();
	int main()
	{
	while(i)
	{
	main();
	goodday();
	i++;
	}
	printf("Exam\n");
	return 0;
	}
	int goodday()
	{
	printf("Goodday");
	}

Option A:	Goodday
Option B:	Exam Goodday
Option C:	**Exam**
Option D:	Goodday Exam

1.	Write a program to read Title, Author and Price of 5 books using array of structures. Display the records in ascending order of Price.
2.	Implement a program to perform addition of two matrices.
3.	Write a program to check whether a word is palindrome or not..
4.	What are bitwise and logical operators in C ?
5.	What are strings and give any four string related functions.
6.	Implement a program to find transpose of a matrix.
7.	Write a C program to find LCM of two numbers using recursion.
8.	Distinguish between structure and union.
9.	What are the tokens of c language explain with example.
10.	Explain while loop with example.
11.	Write a program to print Fibonacci series.
12.	Write a program using recursion to find factorial of a number.
13.	Explain nested structures with examples.
14.	Write a C program to perform multiplication of two matrices.
15.	Explain conditional operator used in C language with proper example.
16.	Explain the term recursion. Write a program to find the power of x raised to n that is: x^n, using recursive function.
17.	Explain following functions with example sqrt(), fabs(), pow(), ceil(), floor()
18.	Write a program to print the following pattern. A B B C C C D D D D
19.	Write a program to find largest element of an 1D array.
20.	Write a Program to calculate and display sum of all the elements of the matrix.
21.	Define a structure called player with data members as player name, team name, batting average. Store and display the information of at least 10 players.
22.	Write a program to accept three numbers from the user and display the greatest of three using the conditional operator.
23.	Write a program to display the following for the user specified number of lines. * ** *** **** ***** ******
24.	Write a program to check if the entered number is prime number or not.
25.	Write a program in C to find out the power of x raised to n (xn), using non-recursive

	function.
26.	Write a program in C to find the smallest of N elements using an array.
27.	Write a program in C to find the reverse of a given string without using inbuilt string function.
28.	Write a program to accept a set of 10 numbers and print the numbers using arrays. Find the average of these integers.
29.	Write a program to store and display at least 10 records of the name, roll number and fees of a student using structure.
30.	Explain five arithmetic operators used in C language with proper examples.
31.	Explain String function for the following operations with example. i) Copy string from source to destination. ii) Merging of two strings.
32.	Explain the term recursion. Write a program to find summation of n numbers using recursion.
33.	Write a program to print the following pattern. (Note- Not only 4 lines, it should print N lines taken from the user.) A B B C C C D D D D
34.	Write a C-program to create array of structures in order to store details of almost 100 books. The book details are book name, book price, book page number and book author name.
35.	Write a program that will accept two-dimensional square matrix and find the sum of diagonal elements. (Note- sum of diagonal elements should be calculated for both sides).
36.	Explain the use of following in-built functions of C-language by giving suitable programming examples and also mention their respective header files in which they are defined. i) getch() ii) pow() iii) ceil() iv) puts() v) getchar()
37.	What are the different ways of parameter passing to a function? Explain with examples.
38.	Write a C program to find GCD of two numbers using recursion.
39.	Write a C program to implement month name by accepting month number from user. (Use switch case)
40.	Write a C program to accept 10 integers from the user and arrange them in ascending order and display them.
41.	Give the difference between entry and exit controlled loop with an example.
42.	Differentiate between arrays and structures.

1 : Write a program to read Title, Author and Price of 5 books using array of structures. Display the records in ascending order of Price.*/

```
#include<conio.h>
#include<stdio.h>
struct book
{
 char title[20];
 struct
 {
  char f_name[20],surname[20];
  }author;
 float price;
};
void main ()
{
struct book s[10],temp;
int n=5,i,j;
float x;
clrscr();
for(i=0;i<=n-1;i++)
{
printf("Enter the title, author's name and price:");
scanf("%s %s %s %f",s[i].title,
s[i].author.f_name,s[i].author.surname,&x);
s[i].price=x;
}
for(i=0;i<=n-1;i++)
{
for(j=0;j<=n-2;j++)
{
if(s[j].price>s[j+1].price)
{
temp=s[j];
s[j]=s[j+1];
s[j+1]=temp;
}
}
}
printf("Title\tAuthor\tPrice\n");
printf("--------------------------------\n");
for(i=0;i<=n-1;i++)
{
printf("%s\t%s\t%s\t%f\n",s[i].title,s[i].author.f_name, s[i].author.surname,s[i].price);
```

```
  }
  getch();
  }
```

2. Implement a program to perform addition of two matrices.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int m,n,i,j,a[10][10],b[10][10],c[10][10];
clrscr();
printf("Enter the number of rows and columns:");
scanf("%d %d",&m,&n);
printf("Enter the elements of Matrix 1\n");
for(i=0;i<=m-1;i++)
{
for(j=0;j<=n-1;j++)
{
printf("Enter a value:");
scanf("%d",&a[i][j]);
}
}
printf("Enter the elements of Matrix 2\n");
for(i=0;i<=m-1;i++)
{
for(j=0;j<=n-1;j++)
{
printf("Enter a value:");
scanf("%d",&b[i][j]);
}
}
for(i=0;i<=m-1;i++)
{
for(j=0;j<=n-1;j++)
{
c[i][j]=a[i][j]+b[i][j];
}
}
printf("The sum of two matrices is:\n");
for(i=0;i<=m-1;i++)
{
for(j=0;j<=n-1;j++)
{
```

```
printf("%d\t",c[i][j]);
}
printf("\n");
}
getch();
}
```

3. Write a program to check whether the entered word is palindrome or not.

Note : Palindrome is a string which is same when read from either of the sides. For e.g. nitin, madam, malayalam etc.*/

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n=0,i;
char a[100],rev[100];
clrscr();
printf("Enter a string:");
gets(a);
while(a[n]!='\0')
{
n++;
}
for(i=0;i<=(n-1);i++)
{
rev[n-i-1]=a[i];
}
for(i=0;i<=n-1;i++)
{
if(a[i]!=rev[i])
break;
}
if(i==n)
printf("The string is palindrome.");
else
printf("The string is not palindrome.");
getch();
}
```

6. Implement a program to find transpose of a matrix.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int m,n,i,j,a[10][10];
void transpose (int a[10][10],int m, int n);
clrscr();
printf("Enter the number of rows and columns:");
scanf("%d %d",&m,&n);
for(i=0;i<=m-1;i++)
{
for(j=0;j<=n-1;j++)
{
printf("Enter a value:");
scanf("%d",&a[i][j]);
}
}
printf("The original Matrix is:\n");
for(i=0;i<=m-1;i++)
{
for(j=0;j<=n-1;j++)
{
printf("%d \t",a[i][j]);
}
printf("\n");
}
transpose(a,m,n);
getch();
}
void transpose (int a[10][10], int m, int n)
{
int b[10][10],i,j;
for(i=0;i<=m-1;i++)
{
for(j=0;j<=n-1;j++)
{
b[j][i]=a[i][j];
}
}
printf("The transpose of this matrix is:\n");
for(i=0;i<=n-1;i++)
```

```
{
for(j=0;j<=m-1;j++)
{
printf("%d\t",b[i][j]);
}
printf("\n");
}
}
```

7. Write a C program to find LCM of two numbers using recursion.

```
#include <stdio.h>
int lcm(int a, int b);
void main()
{
   int num1, num2, LCM;

   printf("Enter any two numbers to find lcm: ");
   scanf("%d%d", &num1, &num2);

   if(num1 > num2)
      LCM = lcm(num2, num1);
   else
      LCM = lcm(num1, num2);

   printf("LCM of %d and %d = %d", num1, num2, LCM);
  getch();
}
int lcm(int a, int b)
{
   static int multiple = 0;
   multiple += b;

   if((multiple % a == 0) && (multiple % b == 0))
   {
      return multiple;
   }
   else
   {
      return lcm(a, b);
   }
}
```

11. Write a program to print Fibonacci series.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a=0,b=1,c,i,n;
clrscr();
printf("Enter a number: ");
scanf("%d",&n);
printf("Fibonacci Series\n0\n1\n");
for(i=1;i<=n-2;i++)
{
c=a+b;
printf("%d\n",c);
a=b;
b=c;
}
getch();
}
```

12. Write a program using recursion to find factorial of a number.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int no,factorial;
int fact (int no);
clrscr();
printf("Enter a number:");
scanf("%d",&no);
factorial=fact(no);
printf("Factorial=%d",factorial);
getch();
}
int fact (int no)
{
if(no==1)
return 1;
else
return (no * fact (no-1));
}
```

14.Write a C program to perform multiplication of two matrices.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
clrscr();
printf("Enter the number of rows=");
scanf("%d",&r);
printf("Enter the number of columns=");
scanf("%d",&c);
printf("Enter the first matrix elements=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("Enter the second matrix elements=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
scanf("%d",&b[i][j]);
}
}

printf("Multiplication of the matrices=\n");
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
mul[i][j]=0;
for(k=0;k<c;k++)
{
mul[i][j]+=a[i][k]*b[k][j];
}
}
}
//for printing result
for(i=0;i<r;i++)
```

```
{
for(j=0;j<c;j++)
{
printf("%d\t",mul[i][j]);
}
printf("\n");
}
getch();
}
```

16. Explain the term recursion. Write a program to find the power of x raised to n that is: x^n, using recursive function.

Recursion is the process in which a function calls itself up to n-number of times. If a program allows the user to call a function inside the same function recursively, the procedure is called a recursive call of the function. Furthermore, a recursive function can call itself directly or indirectly in the same program.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,x,y;
int exponential (int x, int n);
clrscr();
printf("Enter the values of x and n:");
scanf("%d %d",&x,&n);
y=exponential(x,n);
printf("The value of x raise to n is %d",y);
getch();
}
int exponential (int x, int n)
{
if(n==1)
return x;
else
return (x*exponential(x,n-1));
}
```

18.Write a program to print the following pattern.

A
B B
C C C
D D D D

```
#include<stdio.h>
void main()
{
int i,j;
char x='A';
for(i=1;i<=4;i++)
{
      for(j=1;j<=i;j++)
      {
        printf("%c",x);
      }
      x++;
      printf("\n");
}
}
```

19. Write a program to find largest element of an 1D array.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
int n,i,a[100],large;
clrscr();
printf("Enter the number of elements:");
scanf("%d",&n);
for(i=0;i<=n-1;i++)
{
printf("Enter a value:");
scanf("%d",&a[i]);
}
large=a[0];
for(i=1;i<=n-1;i++)
{
if(large<a[i])
large=a[i];
}
printf("The largest number is %d",large);
getch();
}
```

20. Write a Program to calculate and display sum of all the elements of the matrix.

```
#include<stdio.h>
void main( )
{
int a[10][10],sum=0,i,j,r,c;
printf("Enter no of rows and columns : ");
scanf("%d%d",&r,&c);
printf("\nEnter elements of the array : ");
for(i=0;i<r;i++)
{
   for(j=0;j<c;j++)
   {
      scanf("%d",&a[i][j]);
   }
}

for(i=0;i<r;i++)
{
   for(j=0;j<c;j++)
   {
      sum=sum+a[i][j];
   }
}
printf("\n\n Sum of all elements of the array = %d",sum);
getch();
}
```

21. Define a structure called player with data members as player name, team name, batting average. Store and display the information of at least 10 players.

```
#include<stdio.h>
#include<conio.h>
struct player
{
 char p_name[25], t_name[25];
 float b_avg;
};

void main()
{
struct player p[10];
int n,i;
clrscr();
```

```
printf("Enter the number of players : ");
scanf("%d",&n);
for(i=0;i<=n-1;i++)
{
 printf("\n Enter the player's name, batting average, and team : ");
 scanf("%s%f%s",p[i].p_name,&p[i].b_avg,p[i].t_name);
}
printf("\nPlayer Name \t Team \t Average \n");
printf("----------------------------\n");
for(i=0;i<=n-1;i++)
{
 printf("%s \t\t %s \t %f \n",p[i].p_name,p[i].t_name,p[i].b_avg);
}
getch();
}
```

22.Write a program to accept three numbers from the user and display the greatest of three using the conditional operator.

```
#include <stdio.h>
#include<conio.h>>
void main()
{
    int a,b,c,greatest;
    clrscr();
    printf("Enter three numbers : ");
    scanf("%d%d%d",&a,&b,&c);
    greatest = (a>b)?((a>c)?a:c):((b>c)?b:c);
    printf("Greatest of three = %d",greatest);
   getch();
}
```

23.Write a program to display the following for the user specified number of lines.

```
      *
     **
    ***
   ****
  *****
******
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,n;
clrscr();
```

```
printf("Enter the number of lines:");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
for(j=1;j<=n-i;j++)
{
printf(" ");
}
for(j=1;j<=i;j++)
{
printf("*");
}
printf("\n");
}
getch();
}
```

24.Write a program to check if the entered number is prime number or not.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,i,m=0,flag=0;
clrscr();
printf("Enter the number to check prime:");
scanf("%d",&n);
m=n/2;
for(i=2;i<=m;i++)
{
if(n%i==0)
{
printf("Number is not prime");
flag=1;
break;
}
}
if(flag==0)
printf("Number is prime");
getch();
}
```

25.Write a program in C to find out the power of x raised to n (x^n), using non-recursive function.

```
#include <stdio.h>
```

```
#include <conio.h>

int Expo(int X, int Y)
{
int power = 1, i;
for (i = 1; i <= Y; ++i)
{
power = power * X;
}
return power;
}
void main()
{
long long int base, exponent;
clrscr();
printf("Enter Base: ");
scanf("%d", &base);
printf("Enter Power: ");
scanf("%d", &exponent);
printf("%d ^ %d = %d", base, exponent, Expo(base, exponent));
getch();
}
```

26.Write a program in C to find the smallest of N elements using an array.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
int n,i,a[100],small;
clrscr();
printf("Enter the number of elements:");
scanf("%d",&n);
for(i=0;i<=n-1;i++)
{
printf("Enter a value:");
scanf("%d",&a[i]);
}
small=a[0];
for(i=1;i<=n-1;i++)
{
if(small>a[i])
small=a[i];
}
```

```
printf("The smallest number is %d",small);
getch();
}
```

27. Write a program in C to find the reverse of a given string without using inbuilt string function.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n=0,i;
char a[100],temp;
clrscr();
printf("Enter a string:");
gets(a);
while(a[n]!='\0')
{
n++;
}
for(i=0;i<=(n-1)/2;i++)
{
temp=a[n-i-1];
a[n-i-1]=a[i];
a[i]=temp;
}
printf("The reverse of this string is: %s",a);
getch();
}
```

28.Write a program to accept a set of 10 numbers and print the numbers using arrays. Find the average of these integers.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n=10,i,a[10],sum=0;
float avg;
clrscr();
printf("Enter the 10 numbers:");

for(i=0;i<=n-1;i++)
{
printf("\nEnter number %d:",i+1);
```

```
scanf("%d",&a[i]);
}
printf("\nEntered numbers are:\n");
for(i=0;i<=n-1;i++)
{
printf("\nNumber %d:%d",i+1,a[i]);
}
for(i=0;i<=n-1;i++)
{
sum=sum+a[i];
}
avg=sum;
avg=avg/n;
printf("\n\nThe average of the numbers entered is %.2f",avg);
getch();
}
```

29. Write a program to store and display at least 10 records of the name, roll number and fees of a student using structure.

```
#include<stdio.h>
#include<conio.h>
struct student
{
 char name[25];
 int roll_no,fees;
};

void main()
{
struct student s[10];
int n,i;
clrscr();
printf("Enter the number of students : ");
scanf("%d",&n);
for(i=0;i<=n-1;i++)
{
 printf("\n Enter the student's name, roll number, and fees: ");
 scanf("%s%d%d",s[i].name,&s[i].roll_no,&s[i].fees);
}
printf("\nStudent Name \t Roll Number \t Fees \n");
printf("----------------------------\n");
for(i=0;i<=n-1;i++)
```

```
{
 printf("%s \t\t %d \t %d \n", s[i].name,s[i].roll_no,s[i].fees);
}
getch();
}
```

32. Explain the term recursion. Write a program to find summation of n numbers using recursion.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int no,sum;
int summation (int no);
clrscr();
printf("Enter a number:");
scanf("%d",&no);
sum=summation(no);
printf("Sum of numbers from 1 to %d is %d",no,sum);
getch();
}
int summation (int no)
{
if(no==1)
return 1;
else
return (no + summation (no-1));
}
```

33.Write a program to print the following pattern. (Note- Not only 4 lines, it should print N lines taken from the user.)

```
        A
      B   B
    C   C   C
  D   D   D   D
```

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,n;
char x='A';
clrscr();
printf("Enter the number of lines:");
```

```
scanf("%d",&n);
for(i=1;i<=n;i++)
{
for(j=1;j<=n-i;j++)
{
printf(" ");
}
for(j=1;j<=i;j++)
{
printf("%c ",x);
}
x++;
printf("\n");
}
getch();
}
```

34. Write a C-program to create array of structures in order to store details of almost 100 books. The book details are book name, book price, book page number and book author name.

```
#include<stdio.h>
#include<conio.h>
struct book
{
 char name[25],author[25];
 int price,p_no;
};

void main()
{
struct book b[100];
int n,i;
clrscr();
printf("Enter the number of books : ");
scanf("%d",&n);
for(i=0;i<=n-1;i++)
{
 printf("\n Enter the book's name, price, page number, and author name: ");
 scanf("%s%d%d%s",b[i].name,&b[i].price,&b[i].p_no,b[i].author);
}
printf("\nBook Name \t Price \t Page Number \t Author Name \n");
printf("-------------------------------------------------\n");
for(i=0;i<=n-1;i++)
```

```
{
 printf("%s  \t  %d  \t  %d  \t %s \n", b[i].name,b[i].price,b[i].p_no,b[i].author);
}
getch();
}
```

35. Write a program that will accept two-dimensional square matrix and find the sum of diagonal elements. (Note- sum of diagonal elements should be calculated for both sides).

```
#include<stdio.h>
#include<conio.h>
void main()
{
int m,n,i,j,a[10][10],sum=0;
clrscr();
printf("Enter the number of rows / columns:");
scanf("%d",&m);
n=m;
for(i=0;i<=m-1;i++)
{
for(j=0;j<=n-1;j++)
{
printf("Enter a value:");
scanf("%d",&a[i][j]);
}
}
printf("\nEntered matrix is: \n");
for(i=0;i<=m-1;i++)
{
for(j=0;j<=n-1;j++)
{
printf("%d\t",a[i][j]);
}
printf("\n");
}
for(i=0;i<=m-1;i++)
{
for(j=0;j<=n-1;j++)
{
if(i==j)
sum+=a[i][j];
}
}
```

```
printf("The sum of diagonal elements is %d",sum);
getch();
}
```

38. Write a C program to find GCD of two numbers using recursion.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int no1,no2,result;
int GCD (int no1, int no2);
clrscr();
printf("Enter two numbers:");
scanf("%d %d",&no1,&no2);
result=GCD(no1,no2);
if(result==1)
printf("GCD doesnt exist");
else
printf("GCD=%d",result);
getch();
}
int GCD (int no1, int no2)
{
if(no1%no2==0)
return no2;
else
return(GCD(no2,no1%no2));
}
```

39. Write a C program to implement month name by accepting month number from user. (Use switch case)

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int choice;
   clrscr();
   printf("\n Enter the month number : ");
   scanf("%d",&choice);
   switch(choice)
   {
      case 1: printf("Month 1 is :  January");
            break;
      case 2: printf("Month 2 is :  February");
            break;
```

```
        case 3: printf("Month 3 is :  March");
                break;
        case 4: printf("Month 4 is :  April");
                break;
        case 5: printf("Month 5 is :  May");
                break;
        case 6: printf("Month 6 is :  June");
                break;
        case 7: printf("Month 7 is :  July");
                break;
        case 8: printf("Month 8 is :  August");
                break;
        case 9: printf("Month 9 is :  September");
                break;
        case 10: printf("Month 10 is :  October");
                break;
        case 11: printf("Month 11 is :  November");
                break;
        case 12: printf("Month 12 is :  December");
                break;
        default : printf("Invalid month number. Enter between 1 to 12");
    }
getch();
}
```

40. Write a C program to accept 10 integers from the user and arrange them in ascending order and display them.

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int element[10],i,j,temp;
  printf("Enter 10 integer numbers:");
  for(i=0;i<10;i++)
 {
     scanf("%d",&element[i]);
  }

for(i=0;i<10-1;i++)
    {
     for(j=i+1;j<10;j++)
    {
        if(element[i]>element[j])
       {
          temp=element[i]; //swapping element[i] with element[j]
          element[i]=element[j];
          element[j]=temp;
        }
```

```
    }
  }
  printf("Elements in ascending order:\n");
  for(i=0;i<10;i++)
    printf("%d\n",element[i]);
  getch();
}
```

Differentiate between arrays and structures

BASIS FOR COMPARISON	ARRAY	STRUCTURE
Basic	An array is a collection of variables of same data type.	A structure is a collection of variables of different data type.
Syntax	type array_name[size];	struct sruct_name{ type element1; type element1; . . } variable1, variable2, . .;
Memory	Array elements are stored in contiguous memory location.	Structure elements may not be stored in a contiguous memory location.
Access	Array elements are accessed by their index	Structure elements are accessed by their names.

BASIS FOR COMPARISON	ARRAY	STRUCTURE
	number.	
Operator	Array declaration and element accessing operator is "[ ]" (square bracket).	Structure element accessing operator is "." (Dot operator).
Pointer	Array name points to the first element in that array so, array name is a pointer.	Structure name does not point to the first element in that structure so, structure name is not a pointer.
Objects	Objects (instances) of an array can not be created.	Structure objects (instance or structure variable) can be created.
Size	Every element in array is of same size.	Every element in a structure is of different data type.
Bit filed	Bit filed can not be defined in an array.	Bit field can be defined in a structure.
Keyword	There is no keyword to declare an array.	"struct" is a keyword used to declare the structure.

BASIS FOR COMPARISON	ARRAY	STRUCTURE
User-defined	Arrays are not user-defined they are directly declared.	Structure is a user-defined datatype.
Accessing	Accessing array element requires less time.	Accessing a structure elements require comparatively more time.
Searching	Searching an array element takes less time.	Searching a structure element takes comparatively more time than an array element.

5. What are the different ways of parameter passing to a function? Explain with examples

Parameter Passing Methods

- Call by value
- Call by reference

Call by value

- Actual argument passed to the formal argument.
- Any changes to the formal argument does not affect the actual argument

```
#include <stdio.h>
#include<conio.h>
int add(int,int);
int main()
{
        int a,b,c;
        clrscr();
        printf("\nEnter two number:");
        scanf("%d%d",&a,&b);
        c=add(a,b);
        printf("\nSum is:%d",c);
}
int add(int x,int y)
{
 int z;
 z=x+y;
```

```
 return(z);
}
```

Call by reference

- Instead of passing value, the address of the argument will be passed.
- Any changes to the formal argument will affect the actual argument.

```
#include <stdio.h>
#include<conio.h>
void swap(int*,int*);
void main()
{
        int x,y;
        printf("\nEnter value of x:");
        scanf("%d",&x);
        printf("\nEnter value of y:");
        scanf("%d",&y);
swap(&x,&y);
        printf("\nx=%d,y=%d",x,y);
}
void swap(int *a,int *b)
{
        int c;
        c=*a;
        *a=*b;
        *b=c;
 printf("\nx=%d,y=%d",*a,*b);
}
```

6.Explain the use of following in-built functions of C-language by giving suitable programming examples and also mention their respective header files in which they are defined

vi) getch()
vii) pow()
viii) ceil()
ix) puts()
x) getchar()

Ans: 1) getch() : The getch() is **a predefined non-standard function that is defined in conio. h header file**. It is mostly used by the Dev C/C++, MS- DOS's compilers like Turbo C to hold the screen until the user passes a single value to exit from the console screen.

```
// Example for getch() in C
#include <stdio.h>
// Library where getch() is stored
#include <conio.h>
int main()
{
printf("%c", getch());
return 0;
}
```

ii) pow()

iii) The pow() function computes the power of a number.

iv) The pow() function takes two arguments (base value and power value) and, returns the power raised to the base number. For example,

```
v)   [Mathematics] xʸ = pow(x, y) [In programming]
vi)  #include <stdio.h>
vii) #include <math.h>
viii)
ix)  int main()
x)   {
xi)      double base, power, result;
xii)
xiii)     printf("Enter the base number: ");
xiv)     scanf("%lf", &base);
xv)
xvi)     printf("Enter the power raised: ");
xvii)     scanf("%lf",&power);
xviii)
xix)     result = pow(base,power);
xx)
xxi)     printf("%.1lf^%.1lf = %.2lf", base, power, result);
xxii)
xxiii)    return 0;
xxiv) }
```

xi) ceil()

xii) The ceil() function computes the nearest integer greater than the argument passed.

xiii) C ceil() Prototype

```
xiv) double ceil( double arg );

xv)  #include <stdio.h>
xvi) #include <math.h>
xvii)
xviii)int main()
xix) {
xx)     double num = 8.33;
xxi)    int result;
xxii)
xxiii)   result = ceil(num);
xxiv)    printf("Ceiling integer of %.2f = %d", num, result);
xxv)
xxvi)    return 0;
xxvii)}
Output
```

```
Ceiling integer of 8.33 = 9
```

iii) puts()

The **puts function** in C is used to write a line or string to the output stream (**stdout**) that is up to, but does not include, the null character. The puts function also appends a newline character to the output and returns an integer.

To use the puts function, you need to include the **<stdio.h>** library in the program. This is shown below:

```c
#include <stdio.h>
#include <string.h>

int main() {

  // initializing strings
  char str1[] = "Hello World";
  char str2[] = "Using puts in C";

  // writing to stdout
  puts(str1);
  puts(str2);

  return 0;
}
```

iv) getchar()

The `getchar` **function** is part of the `<stdio.h>` header file in C. It is used when single character input is required from the user. The function reads the input as an `unsigned char`; then it casts and returns as an `int` or an `EOF`

`EOF` is returned if the end of file is reached or an error is encountered.

```c
#include<stdio.h> // Including header file

int main(){

  int myChar; // creating a variable to store the input
  myChar = getchar(); // use getchar to fetch input
  printf("You entered:  %c", myChar); // print input on screen
  return 0;
}
```