

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

	<pre> int main() { int a=5,i; i!=a >10; printf(“i=%d”,i) ;return 0; } </pre>
Option A:	i=0
Option B:	i=10
Option C:	i=110
Option D:	i=1
5.	<p>How many times will the following while-loop repeat, i.e., how many x are printed?</p> <pre> int main() { int i = 5; while(i>0) { printf(“x”) ;i--; } return 0; } </pre>
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. <pre>int main() { int a=0; a = 5<2 ? 4 : 3; printf("%d", a);return 0; }</pre>
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	%s

A:	
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?

	<pre>#include <stdio.h>void main() { int x = 5; if (x == 5) printf("hi\n");else printf("how are u\n"); printf("hello\n"); }</pre>
Option A:	hi
Option B:	hi hel lo
Option C:	how are youhello
Option D:	how are you
20	<p>What will be the output of the following C code? (Assuming that we have entered the value 1 in the standard input).</p> <pre>#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"); } }</pre>
Option A:	1
Option B:	1

	hi
Option C:	hi
Option D:	2
21	<p>What will be the output of the following C code?</p> <pre>#include <stdio.h>int main() { int i = 0; while (i = 0) printf("True\n"); printf("False\n"); }</pre>
Option A:	True
Option B:	False
Option C:	True False
Option D:	True (Infinite Times)

22	<p>What will be the output of the following C code?</p> <pre>#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"); }</pre>

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

D:	
29	What will be the output of the following program? <pre>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; }</pre>
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? <pre>while((a>0)&&(a<25)) { loopbod y a++; }</pre>
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? <pre>int main() { int a = 500, b = 100, c;if(!a >= 400) b = 300; else</pre>

	<pre> b=b+++b*a/b ;c = 10; c=b<<1; c=c>>b+1; printf("b = %d c = %d\n", b, c);return 0; } </pre>
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? <pre> int i; int goodday();int main() { while(i) { main(); goodday() ;i++; } printf("Exam\n") ;return 0; } int goodday() { printf("Goodday"); } </pre>
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.

Program :

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

```
//#include <conio.h>
```

```
struct book
```

```
{
```

```
int price;
```

```
char title[80];
```

```
char author[80];
```

```
};
```

```
void accept(struct book list[80]);
```

```
//func declare
```

```
void display(struct book list[80]);
```

```
void bsortAsc(struct book list[80]);
```

```
void main()
```

```
{
```

```
struct book data[20];
```

```
int n;
```

```
//clrscr();
```

```
accept(data);
```

```
//func call
```

```
bsortAsc(data);
```

```
display(data);
```

```
//getch();
```

```
}
```

```
void accept(struct book list[5])
```

```
// func initialize
```

```
{
```

```
int i;
```

```
for (i = 0; i < 5; i++)
```

```
{
```

```
printf("\nEnter title : ");
```

```
scanf("%s", &list[i].title);
```

```
printf("Enter Author name: ");
```

```
scanf("%s",&list[i].author);
```

```
printf("Enter price : ");
```

```
scanf("%d", &list[i].price);
```

```
}
```

	<pre> } void display(struct book list[80]) { int i; printf("\n\nTitle\t\tAuthor\t\tprice\n"); printf("-----\n"); for (i = 0; i<5; i++) { printf("%s\t\t%s\t\t%d\n", list[i].title, list[i].author, list[i].price); } } void bsortAsc(struct book list[80]) { int i, j; struct book temp; for (i = 0; i < 5 ; i++) { for (j = 0; j < (5 -i); j++) { if (list[j].price >list[j + 1].price) { temp = list[j]; list[j] = list[j + 1]; list[j + 1] = temp; } } } } </pre>
2.	<p>Implement a program to perform addition of two matrices.</p> <p>Program:</p> <pre> #include <stdio.h> int main() { int row,col, i, j, first[10][10], second[10][10], sum[10][10]; printf("Enter the number of rows and columns of matrix\n"); scanf("%d %d", &row, &col); </pre>

	<pre> printf("Enter the elements of first matrix\n"); for (i = 0; i < row; i++) for (j = 0; j < col; j++) scanf("%d", &first[i][j]); printf("Enter the elements of second matrix\n"); for (i = 0; i < row; i++) for (j = 0; j < col; j++) scanf("%d", &second[i][j]); printf("Sum of entered matrices:-\n"); for (i = 0; i < row; i++) { for (j = 0; j < col; j++) { sum[i][j] = first[i][j] + second[i][j]; printf("%d\t", sum[i][j]); } printf("\n"); } return 0; } </pre>
3.	<p>Write a program to check whether a word is palindrome or not.</p> <p>Program :</p> <pre> #include<stdio.h> #include<conio.h> #include<string.h> int main(){ char str[20]; int i, len, temp=0; int flag = 0; printf("Enter a string:"); scanf("%s", str); len = strlen(str); for(i=0;i < len ;i++){ if(str[i] != str[len-i-1]){ temp = 1; break; } } </pre>

	<pre> } if (temp==0) { printf("String is a palindrome"); } else { printf("String is not a palindrome"); } return 0; } </pre>
<p>4.</p> <p>Ans</p> <p>.</p>	<p>What are bitwise and logical operators in C ?</p> <p>1.Bitwise operators: Perform operations on individual bits, and the result is also always a bit.This type of operator has two types of variables it can work with. One is the integral numeric type,There are a total of six bitwise operators: ~ - Complement, & - AND, - OR, ^ - Exclusive OR ,<< - Left shift , >> - Right shift</p> <p>2.Logical operators: Compare bits of the given object and always return a Boolean result.This category of operators works with Boolean variables or expressions. There are three basic operands it allows you to use: && (AND), (OR), and ! (NOT).</p>
<p>5.</p> <p>Ans</p> <p>.</p>	<p>What are strings and give any four string related functions.</p> <p>1)A string is a one-dimensional array of characters terminated a null character('\0').Strings are enclosed by double quotes e.g. "name".</p> <p>Syntax:</p> <pre> #include<stdio.h> int main (){ char str1[20]={'H','e','l','l','o','\0'}; char str2[20]="world"; printf ("%s ", str1); printf ("%s ", str2); return 0; } </pre> <p>2)For using the string functions include <string.h> header file.</p> <p>a)strlen()</p> <p>This function returns an integer value that is the length of the string passed to the function.</p> <pre> #include<conio.h> </pre>

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

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

```
void main()
```

```
{
```

```
    int l;
```

```
    char a[100];
```

```
    clrscr();
```

```
    printf("enter a string\n");
```

```
    gets(a);
```

```
    l=strlen(a);
```

```
    printf("the length of the entered string is: %d",l);
```

```
    getch();
```

```
}
```

```
b)strcpy()
```

This function copies the second string into the first string passed to the function. The second string remains unchanged.

```
#include<conio.h>
```

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

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

```
void main()
```

```
{
```

```
    char a[100],b[100];
```

```
    clrscr();
```

```
    printf("enter a string\n");
```

```
    gets(a);
```

```
    strcpy(b,a);
```

```
    printf("the new string is %s",b);
```

```
    getch();
```

```
}
```

```
c)Strcmp()
```

This function compares the two string variables passed to it. it returns an integer value equal to:

0(zero), if the two strings are equal.

Negative value, if the first string is smaller than the second string.

Positive value, if the first string is greater than the second string.

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

```
#include<conio.h>
```

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


	<pre> void main() { char a[100],b[100]; clrscr(); printf("enter two strings:\n"); gets(a); gets(b); if(strcmp(a,b)==0) printf("strings are equal "); else printf("%s string is greater ",a); else printf("%s string is greater",b); getch(); } </pre> <p>d)strcat()</p> <p>This function concatenates (joins) the two string variables passed to it.It returns a string of the combination of the two in the first string variable.</p> <pre> #include<conio.h> #include<stdio.h> #include<string.h> void main() { char a[100],b[100]; clrscr(); printf("enter two strings:\n"); gets(a); gets(b); strcat(a,b); printf("the concatenated strig is %s",a); getch(); } </pre>
6.	<p>Implement a program to find transpose of a matrix.</p> <p>Program :</p> <pre> #include <stdio.h> int main() { int a[10][10], transpose[10][10], r, c; printf("Enter rows and columns: "); scanf("%d %d", &r, &c); </pre>

	<pre> // assigning elements to the matrix printf("\nEnter matrix elements:\n"); for (int i = 0; i < r; ++i) for (int j = 0; j < c; ++j) { printf("Enter element a%d%d: ", i + 1, j + 1); scanf("%d", &a[i][j]); } // printing the matrix a[][] printf("\nEnter matrix: \n"); for (int i = 0; i < r; ++i) for (int j = 0; j < c; ++j) { printf("%d ", a[i][j]); if (j == c - 1) printf("\n"); } // computing the transpose for (int i = 0; i < r; ++i) for (int j = 0; j < c; ++j) { transpose[j][i] = a[i][j]; } // printing the transpose printf("\nTranspose of the matrix:\n"); for (int i = 0; i < c; ++i) for (int j = 0; j < r; ++j) { printf("%d ", transpose[i][j]); if (j == r - 1) printf("\n"); } return 0; } </pre>
7.	<p>Write a C program to find LCM of two numbers using recursion.</p> <p>Program:</p> <pre>#include<stdio.h></pre>

	<pre> #include<conio.h> int lcm(int a , int b) { static int m = 0; m = m + b; if(m % a == 0 && m % b == 0) { return m; } return lcm(a , b); } int main() { int x , y; printf("Enter First Number : "); scanf("%d" , &x); printf("Enter Seconf Number : "); scanf("%d" , &y); printf("LCM of %d and %d = %d" , x , y , lcm(x , y)); return 0; } Output: Enter First Number : 36 Enter Second Number : 20 LCM of 36 and 20 = 180 </pre>
8.	Distinguish between structure and union.

		STRUCTURE	UNION
	Keyword	The keyword struct is used to define a structure	The keyword union is used to define a union.
	Size	When a variable is associated with a structure, the compiler allocates the memory for each member. The size of structure is greater than or equal to the sum of sizes of its members.	when a variable is associated with a union, the compiler allocates the memory by considering the size of the largest memory. So, size of union is equal to the size of largest member.
	Memory	Each member within a structure is assigned unique storage area of location.	Memory allocated is shared by individual members of union.
	Value Altering	Altering the value of a member will not affect other members of the structure.	Altering the value of any of the member will alter other member values.
	Accessing members	Individual member can be accessed at a time.	Only one member can be accessed at a time.
	Initialization of Members	Several members of a structure can initialize at once.	Only the first member of a union can be initialized.
9.	What are the tokens of c language explain with example.		
Ans	<p>Tokens are the smallest elements of a program, which are meaningful to the compiler. Each and every punctuation and word that you come across in a C program is token.</p> <p>The following are the types of tokens: Keywords, Identifiers, Constant, Strings, Operators, etc.</p> <p>Identifiers: These are used to name the arrays, functions, structures, variables, etc. The identifiers are user-defined words in the C language.</p> <p>Keyword: We can define the keywords as the reserved or pre-defined words that hold their own importance. It means that every keyword has a functionality of its own. Eg. auto, break, if, etc.</p> <p>Operators: The operators in C are the special symbols that we use for performing various functions. On the basis of the total number of operands, here is how we classify the operators: Unary Operator, Binary Operator and Ternary Operator.</p> <p>String : A string is a one-dimensional array of characters terminated a null character('\0'). Strings are enclosed by double quotes e.g. "name".</p> <p>Constant : Constant is basically a value of a variable that does not change throughout a program. The constants remain the same, and we cannot change their value whatsoever.</p>		
10.	Explain while loop with example.		
Ans	<p>The while loop evaluates the testExpression inside the parentheses ().</p> <p>If testExpression is true, statements inside the body of while loop are executed. Then, testExpression is evaluated again.</p> <p>The process goes on until testExpression is evaluated to false.</p> <p>If testExpression is false, the loop terminates (ends).</p>		

	<p>The syntax of the while loop is:</p> <pre>while (testExpression) { // the body of the loop }</pre> <p>// Print numbers from 1 to 5</p> <pre>#include <stdio.h> int main() { int i = 1; while (i <= 5) { printf("%d ", i); ++i; } return 0; }</pre> <p>Output: 1 2 3 4 5</p>
11.	<p>Write a program to print Fibonacci series.</p> <p>Program :</p> <pre>#include <stdio.h> #include <conio.h> int main() { int i, n; // initialize first and second terms int t1 = 0, t2 = 1; // initialize the next term (3rd term) int nextTerm = t1 + t2; clrscr(); // get no. of terms from user printf("Enter the number of terms: "); scanf("%d", &n); // print the first two terms t1 and t2 printf("Fibonacci Series: %d, %d, ", t1, t2); // print 3rd to nth terms for (i = 3; i <= n; ++i) { printf("%d ", nextTerm); t1 = t2;</pre>

	<pre> t2 = nextTerm; nextTerm = t1 + t2; } return 0; getch(); } </pre> <p>Output:</p> <p>Enter the number of terms: 10</p> <p>Fibonacci Series: 0 1 1 2 3 5 8 13 21 34</p>
12.	<p>Write a program using recursion to find factorial of a number.</p> <p>Program:</p> <pre> #include<stdio.h> #include<conio.h> int fact(int); int main() { int x,n; clrscr(); printf(" Enter the Number to Find Factorial :"); scanf("%d",&n); x=fact(n); printf(" Factorial of %d is %d",n,x); return 0; getch(); } int fact(int n) { if(n==0) return(1); return(n*fact(n-1)); } </pre> <p>OUTPUT:</p> <p>Enter the Number to Find Factorial :5</p>

	Factorial of 5 is 120
13.	<p>Explain nested structures with examples.</p> <p>C provides us the feature of nesting one structure within another structure by using which, complex data types are created. For example, we may need to store the address of an entity employee in a structure. The attribute address may also have the subparts as street number, city, state, and pin code. Hence, to store the address of the employee, we need to store the address of the employee into a separate structure and nest the structure address into the structure employee. Consider the following program.</p> <pre> #include<stdio.h> #include<conio.h> struct address { char city[20]; int pin; char phone[14]; }; struct employee { char name[20]; struct address add; }; void main () { struct employee emp; clrscr(); printf("Enter employee information?\n"); scanf("%s %s %d %s",emp.name,emp.add.city, &emp.add.pin, emp.add.phone); printf("Printing the employee information....\n"); printf("name: %s\nCity: %s\nPincode: %d\nPhone: %s",emp.name,emp.add.city,emp.add.pin,emp.add.phone); getch(); } </pre> <p>Output:</p> <p>Enter employee information?</p>

	<p>Arun</p> <p>Delhi</p> <p>110001</p> <p>1234567890</p> <p>Printing the employee information....</p> <p>name: Arun</p> <p>City: Delhi</p> <p>Pincode: 110001</p> <p>Phone: 1234567890</p>
14.	<p>Write a C program to perform multiplication of two matrices.</p> <p>Program :</p> <pre> #include<stdio.h> #include<conio.h> int main(){ int a[10][10],b[10][10],mul[10][10],r,c,i,j,k; clrscr(); printf("enter the number of row="); scanf("%d",&r); printf("enter the number of column="); scanf("%d",&c); printf("enter the first matrix element=\n"); for(i=0;i<r;i++) { for(j=0;j<c;j++) { scanf("%d",&a[i][j]); } } printf("enter the second matrix element=\n"); for(i=0;i<r;i++) { for(j=0;j<c;j++) { scanf("%d",&b[i][j]); } } </pre>


```

printf("multiply of the matrix=\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");
}
return 0;
getch();
}

```

Output:

```

enter the number of row=3
enter the number of column=3
enter the first matrix element=
1 1 1
2 2 2
3 3 3
enter the second matrix element=
1 1 1
2 2 2
3 3 3
multiply of the matrix=

```

	6 6 6 12 12 12 18 18 18
15.	<p>Explain conditional operator used in C language with proper example.</p> <p>Conditional operator is also known as ternary operator because it requires three operands and can be used to replace simple if-else code.</p> <p>The syntax of ternary operator is :</p> <p>testCondition ? expression1 : expression 2;</p> <p>The testCondition is a boolean expression that results in either true or false. If the condition is</p> <p>true - expression1 (before the colon) is executed</p> <p>false - expression2 (after the colon) is executed</p> <p>Program:</p> <pre>#include <stdio.h> #include <conio.h> int main() { int age; clrscr(); // take input from users printf("Enter your age: "); scanf("%d", &age); // ternary operator to find if a person can vote or not (age >= 18) ? printf("You can vote") : printf("You cannot vote"); return 0; getch(); }</pre> <p>Output :</p> <p>Enter your age: 12</p> <p>You cannot vote</p>
16. Ans •	<p>Explain the term recursion. Write a program to find the power of x raised to n that is: x^n, using recursive function.</p> <p>A function that calls itself is known as a recursive function. And, this technique is known as recursion. The recursion continues until some condition is met to prevent it.</p>

	<p>To prevent infinite recursion, if...else statement (or similar approach) can be used where one branch makes the recursive call, and other doesn't.</p> <pre> #include <stdio.h> #include <conio.h> int power(int n1, int n2); int main() { int base, a, result; clrscr(); printf("Enter base number: "); scanf("%d", &base); printf("Enter power number(positive integer: "); scanf("%d", &a); result = power(base, a); printf("%d^%d = %d", base, a, result); getch(); return 0; } int power(int base, int a) { if (a != 0) return (base * power(base, a - 1)); else return 1; } </pre> <p>Output:</p> <p>Enter base number: 3</p> <p>Enter power number(positive integer): 4</p> <p>3^4 = 81</p>
<p>17.</p> <p>Ans</p> <p>•</p>	<p>Explain following functions with examples sqrt(), fabs(), pow(), ceil(), floor()</p> <p>All of the functions can be used after including math.h header file.</p>

The sqrt() function computes the square root of a number.

```
#include <math.h>
#include <stdio.h>
int main() {
    double number, squareRoot;
    printf("Enter a number: ");
    scanf("%lf", &number);
    // computing the square root
    squareRoot = sqrt(number);
    printf("Square root of %.2lf =
%.2lf", number, squareRoot);
    return 0;
}
```

Output:

Enter a number: 23.4

Square root of 23.40 = 4.8

The fabs() function returns the absolute value of a number.

```
#include <stdio.h>
#include <math.h>
int main()
{
    double x, result;
    x = -1.5;
    result = fabs(x);
    printf("|%.2lf| = %.2lf\n", x,
result);
    return 0;
}
```

Output:

|-1.50| = 1.50

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

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

```
#include <math.h>
```

```
int main()
```

```
{
```

```
    double base, power, result;
```

```
    printf("Enter the base number: ");
```

```
    scanf("%lf", &base);
```

```
    printf("Enter the power raised: ");
```

```
    scanf("%lf",&power);
```

```
    result = pow(base,power);
```

```
    printf("%.1lf^%.1lf = %.2lf", base,  
power, result);
```

```
    return 0;
```

```
}
```

Output:

Enter the base number: 2.5

Enter the power raised: 3.4

2.5^3.4 = 22.54

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

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

```
#include <math.h>
```

```
int main()
```

```
{
```

```
    double num = 8.33;
```

```
    int result;
```

```
    result = ceil(num);
```

```
    printf("Ceiling integer of %.2f = %d",  
num, result);
```

```
    return 0;
```

```
}
```

	<p>Output:</p> <p>Ceiling integer of 8.33 = 9</p> <p>The floor() function calculates the nearest integer less than the argument passed.</p> <pre>#include <stdio.h> #include <math.h> int main() { double num = 2.33; int result; result = floor(num); printf("Floor integer of %.2f = %d", num, result); return 0; }</pre> <p>Output:</p> <p>Floor integer of 2.33 = 2</p>
18.	<p>Write a program to print the following pattern.A</p> <pre>B B C C C D D D D</pre> <p>Program:</p> <pre>#include<stdio.h> void main() { int i, j; clrscr(); for(i=1;i<=5;i++) { for(j=1;j<=i;j++) { printf("%c ", 'A'-1 + i);</pre>

	<pre> } printf("\n"); } getch(); } </pre>
19.	<p>Write a program to find largest element of an 1D array.</p> <pre> #include <stdio.h> int main(){ int num; int arr[25]; // Asking for input printf("Please enter total no. of elements[1 to 25]: "); scanf("%d", &num); for (int i = 0; i < num; ++i){ printf("%d.Enter the number: ", i + 1); scanf("%d", &arr[i]); } for (int i = 0; i < num; ++i){ if (arr[0] < arr[i]){ arr[0] = arr[i]; } } printf("Largest element of the array is: %d", arr[0]); return 0; } </pre> <p>Output:</p> <p>Please enter total no. of elements[1 to 25]: 5</p> <p>1.Enter the number: 12</p> <p>2.Enter the number: 17</p> <p>3.Enter the number: 5</p> <p>4.Enter the number: 44</p> <p>5.Enter the number: 13</p> <p>Largest element of the array is: 44</p>
20.	<p>Write a Program to calculate and display sum of all the elements of the matrix.</p>

	<pre> #include<stdio.h> int main() { int a[10][10],r,c,sum=0,i,j; printf("/*How Many Rows You Want To \nEnter in Matrix*/\nEnter Limit : "); scanf("%d",&r); printf("\n/*How Many Columns You Want To \nEnter in Matrix*/\nEnter Limit : "); scanf("%d",&c); printf("\nEnter Elements for Matrix of Size %d*%d:\n\n",r,c); for(i=0;i<r;i++){ for(j=0;j<c;j++){ { scanf("%d",&a[i][j]); } printf("\n"); } for(i=0;i<r;i++){ for(j=0;j<c;j++){ sum=sum+a[i][j]; } printf("\nSum of All Elements in Matrix = %d",sum); } return 0; } </pre>
21.	<p>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.</p> <pre> #include<stdio.h> #include<conio.h> struct player { char pname[20]; char tname[20]; float bavg; }; int main() { </pre>

	<pre> struct player s[10]; int i,j,n=10; float p; clrscr(); for(i=0;i<n;i++) { printf("\nEnter PName TName BAvG for player-%d = ",i+1); scanf("%s %s %f",s[i].pname,s[i].tname,&p); s[i].bavg=p; } for(i=0;i<n;i++) { printf("\n%s %s %.2f",s[i].pname,s[i].tname,s[i].bavg); } getch(); return 0; } </pre>
22.	<p>Write a program to accept three numbers from the user and display the greatest of three using the conditional operator.</p> <p>Program:</p> <pre> #include <stdio.h> #include <conio.h> void main() { int a, b, c, big ; clrscr(); printf("Enter three numbers : ") ; scanf("%d %d %d", &a, &b, &c) ; big = a > b ? (a > c ? a : c) : (b > c ? b : c) ; printf("\nThe biggest number is : %d", big) ; getch(); } </pre>
23.	<p>Write a program to display the following for the user specified number of lines.</p> <pre> * ** *** </pre>

	<pre> **** ***** ***** Program: #include <stdio.h> #include <conio.h> int main() { int i, space, rows, k; clrscr(); printf("Enter the number of rows: "); scanf("%d", &rows); printf("\n"); for (i = 1; i <= rows; i++) { for (space = 1; space <= rows - i+1; space++) { printf(" "); } for(k=1; k<=i;k++){ printf("* "); } printf("\n"); } getch(); return 0; } </pre>
24.	<p>Write a program to check if the entered number is prime number or not.</p> <p>Program:</p> <pre> #include <stdio.h> #include <conio.h> void main() { int i, num, p = 0; clrscr(); printf("Please enter a number: \n"); scanf("%d", &num); for(i=1; i<=num; i++) { if(num%i==0) { p++; </pre>

	<pre> } } if(p==2) { printf("Entered number is %d "\ "and it is a prime number.",num); } else { printf("Entered number is %d "\ "and it is not a prime number.",num); } getch(); } </pre> <p>OUTPUT :</p> <p>Please enter a number: 13</p> <p>Entered number is 13 and it is a prime number.</p>
25.	<p>Write a program in C to find out the power of x raised to n (x^n), using non-recursive function.</p> <p>Program:</p> <pre> #include <stdio.h> #include <math.h> int powerres(int base,int power){ return pow(base,power); } int main() { int base, power, result; printf("Enter the base number: "); scanf("%d", &base); printf("Enter the power raised: "); scanf("%d",&power); result = powerres(base,power); printf("%d^%d = %d", base, power, result); </pre>

	<pre> return 0; } Output: Enter the base number: 2 Enter the power raised: 4 2^4 = 16 </pre>
26.	<p>Write a program in C to find the smallest of N elements using an array.</p> <p>Program:</p> <pre> #include<stdio.h> #include<conio.h> void main() { int a[30], i, num, smallest; clrscr(); printf("\nEnter no of elements :"); scanf("%d", &num); //Read n elements in an array for (i = 0; i < num; i++) scanf("%d", &a[i]); //Consider first element as smallest smallest = a[0]; for (i = 0; i < num; i++) { if (a[i] < smallest) { smallest = a[i]; } } // Print out the Result printf("\nSmallest Element : %d", smallest); getch (); } </pre> <p>Output: Enter no of elements : 5 11 44 22 55 99 Smallest Element : 11</p>
27.	<p>Write a program in C to find the reverse of a given string without using inbuilt stringfunction.</p> <p>Program:</p> <pre> #include<stdio.h> #include<string.h> void main() { int i,n; </pre>

	<pre> char str[20]; printf("Enter the String to get reversed: "); gets(str); n=strlen(str); printf("\nReversed string is "); for(i=n-1;i>=0;i--) { printf("%c",str[i]); } } </pre> <p>Output: Enter the String to get reversed: java Reversed string is avaj</p>
28.	Write a program to accept a set of 10 numbers and print the numbers using arrays.

	<p>Find the average of these integers.</p> <p>Program:</p> <pre> #include <stdio.h> #include <conio.h> int main() { int arr[10]; int sum = 0, i; clrscr(); for(i=0; i<10; i++) { printf("Enter number : "); scanf("%d", &arr[i]); } for(i=0; i<10; i++) { printf("%d ",&arr[i]); sum += arr[i]; } printf("Average is %d\n", sum/10); getch(); return 0; } </pre>
29.	Write a program to store and display at least 10 records of the name, roll number and fees of a student using structure.

	<p>Program:</p> <pre> #include<stdio.h> #include<conio.h> struct student { char name[20]; int roll; int fees; }; int main() { struct student s[10]; int i,j,n=10; clrscr(); for(i=0;i<n;i++) { printf("\nEnter Name Roll No. Fees of student-%d = ",i+1); scanf("%s %d %d",s[i].name,s[i].roll,s[i].fees); } for(i=0;i<n;i++) { printf("\n%s %d %d",s[i].name,s[i].roll,s[i].fees); } getch(); return 0; } </pre>				
30.	<p>Explain five arithmetic operators used in C language with proper examples.</p> <p>C Arithmetic Operators</p> <p>An arithmetic operator performs mathematical operations such as addition, subtraction, multiplication, division etc on numerical values (constants and variables).</p> <table> <thead> <tr> <th>Operator</th><th>Meaning of Operator</th></tr> </thead> <tbody> <tr> <td>+</td><td>addition or unary plus</td></tr> </tbody> </table>	Operator	Meaning of Operator	+	addition or unary plus
Operator	Meaning of Operator				
+	addition or unary plus				

	<p>- subtraction or unary minus</p> <p>* multiplication</p> <p>/ division</p> <p>% remainder after division (modulo division)</p> <p>// Working of arithmetic operators</p> <pre>#include <stdio.h> #include <conio.h> int main() { int a = 9,b = 4, c; clrscr(); c = a+b; printf("a+b = %d \n",c); c = a-b; printf("a-b = %d \n",c); c = a*b; printf("a*b = %d \n",c); c = a/b; printf("a/b = %d \n",c); c = a%b; printf("Remainder when a divided by b = %d \n",c); getch(); return 0; }</pre> <p>Output:</p> <p>a+b = 13</p> <p>a-b = 5</p> <p>a*b = 36</p> <p>a/b = 2</p> <p>Remainder when a divided by b=1</p>
31.	<p>Explain String function for the following operations with example.</p> <ul style="list-style-type: none"> Copy string from source to destination. <p>strcpy() takes two strings as arguments and character by character (including \0) copies the content of string Src to string Dest, character by character.</p> <pre>#include <stdio.h></pre>

```

#include <string.h>
#include <conio.h>

int main()
{
    char Src[15]= "DevOps";
    char Dest[15] = "";
    clrscr();
    strcpy(Dest, Src); // calling strcpy function
    printf("After copying\n");
    printf("Source string: %s \n", Src);
    printf("Destination string: %s \n", Dest);
    getch();
    return 0;
}

```

Output

After copying

Source string: DevOps

Destination string: DevOps

- Merging of two strings.

The concatenation of strings is a process of combining two strings to form a single string. If there are two strings, then the second string is appended(added) at the end of the first string.

```

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

int main()
{
    char s1[20]; // declaration of char array
    char s2[20]; // declaration of char array
    printf("Enter the first string : ");
    scanf("%s", &s1);
    printf("\nEnter the second string :");
    scanf("%s",&s2);
    strcat(s1,s2);
    printf("\nThe concatenated string is : %s",s1);
    return 0;
}

```

Output:

Enter the first string : Game

	<p>Enter the second string : Developer The concatenated string is : GameDeveloper</p>
32.	<p>Explain the term recursion. Write a program to find summation of n numbers using recursion.</p> <p>A function that calls itself is known as a recursive function. And, this technique is known as recursion.</p> <p>Program:</p> <pre>#include <stdio.h> #include <conio.h> int addNumbers(int n); int main() { int num; clrscr(); printf("Enter a positive integer: "); scanf("%d", &num); printf("Sum = %d", addNumbers(num)); getch(); return 0; } int addNumbers(int n) { if (n != 0) return n + addNumbers(n - 1); else return n; }</pre> <p>Output: Enter a positive integer: 20 Sum = 210</p>
33.	<p>Write a program to print the following pattern. (Note- Not only 4 lines, it should print N lines taken from the user.)</p> <pre> A B B C C C D D D D </pre> <p>Program:</p>

```

#include<stdio.h>
#include<conio.h>
void main()
{
    int i, j, k, n;
    clrscr();
    printf("Enter any Number:");
    scanf("%d", &n);
    for(i=1; i<=n; i++)
    {
        for(k=n; k>=i; k--)
        printf(" ");
        for(j=1; j<=i; j++)
        {
            printf("%c ", 'A'-1 + i);
        }
        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.

Program :

```

#include <stdio.h>
//#include <conio.h>
struct book
{
    int price;
    char title[80];
    char author[80];
};
void accept(struct book list[100]);           //func declare
void display(struct book list[100]);
void main()
{
    struct book data[20];

```

	<pre> int n; //clrscr(); accept(data); //func call display(data); //getch(); } void accept(struct book list[100]) // func initialize { int i; for (i = 0; i <100; i++) { printf("\nEnter title : "); scanf("%s", &list[i].title); printf("Enter Author name: "); scanf("%s",&list[i].author); printf("Enter price : "); scanf("%d", &list[i].price); } } void display(struct book list[100]) { int i; printf("\n\nTitle\t\tAuthor\t\tprice\n"); printf("-----\n"); for (i = 0; i<100; i++) { printf("%s\t\t%s\t\t%d\n", list[i].title, list[i].author, list[i].price); } } </pre>
35.	<p>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).</p> <p>Program:</p> <pre> #include<stdio.h> #include<conio.h> int main() { int i, j, rows, columns, a[10][10], principal=0, secondary=0 ; </pre>

```

clrscr();
printf("\n Please Enter Number of rows and columns : ");
scanf("%d %d", &i, &j);
printf("\n Please Enter the Matrix Elements \n");
for(rows = 0; rows < i; rows++)
{
    for(columns = 0; columns < j; columns++)
    {
        scanf("%d", &a[rows][columns]);
    }
}
for(rows = 0; rows < i; rows++)
{
    for(columns = 0; columns < j; columns++)
    {
        // Condition for principal diagonal
        if (rows == columns)
            principal += a[rows][columns];

        // Condition for secondary diagonal
        if ((rows+columns) == (i - 1))
            secondary += a[rows][columns];
    }
}
printf("\n The Primary and secondary principal diagonal sum = %d %d",
principal,secondary );

printf("\n The Sum of All Diagonal Elements of a Matrix = %d",
principal+secondary );
getch();
return 0;
}

```

Output:

Please Enter Number of rows and columns : 2 2

Please Enter the Matrix Elements 1 4 5 6

The Primary and secondary principal diagonal sum = 7 9

The Sum of All Diagonal Elements of a Matrix = 16

36.	<p>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.</p> <ul style="list-style-type: none"> getch() <p>getch() is a nonstandard function and is present in conio.h header file. getch() also reads a single character from the keyboard. But it does not use any buffer, so the entered character is immediately returned without waiting for the enter key.</p> <pre> #include <stdio.h> // Library where getch() is stored #include <conio.h> int main() { printf("Hi"); // Now the console will wait // for a key to be pressed getch(); return 0; } Output: Hi </pre> pow() <p>The pow() function computes the power of a number, present in math.h header file.</p> <pre> #include <stdio.h> #include <math.h> int main() { double base, power, result; printf("Enter the base number: "); scanf("%lf", &base); printf("Enter the power raised: "); scanf("%lf", &power); result = pow(base, power); printf("%.1lf^%.1lf = %.2lf", base, power, result); </pre>
-----	--

```
return 0;
}
```

Output:

Enter the base number: 2.5

Enter the power raised: 3.4

2.5^3.4 = 22.54

- **ceil()**

The ceil() function computes the nearest integer greater than the argument passed, present in math.h header file.

```
#include <stdio.h>
#include <math.h>
int main()
{
    double num = 8.33;
    int result;
    result = ceil(num);
    printf("Ceiling integer of %.2f = %d", num,
result);
    return 0;
}
```

Output:

Ceiling integer of 8.33 = 9

- **puts()**

The C library function int puts(const char *str) writes a string to stdout up to but not including the null character. A newline character is appended to the output. Present in string.h header file.

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

int main () {
    char str1[15]="Data Science";
    clrscr();
    puts(str1);
    return(0);
}
```

	<pre> } Output:Data Science • getchar() A getchar() function is a non-standard function whose meaning is already defined in the stdin.h header file to accept a single input from the user. In other words, it is the C library function that gets a single character (unsigned char) from the stdin. #include <stdio.h> #include <conio.h> void main() { char c; clrscr(); printf ("\n Enter a character \n"); c = getchar(); // get a single character printf(" You have passed "); putchar(c); // print a single character using putchar getch(); } Output: Enter a character A You have passed A </pre>
37.	<p>What are the different ways of parameter passing to a function? Explain with examples.</p> <p>Passing Parameter to a Function:</p> <p>In C Programming we have different ways of parameter passing schemes such as Call by Value and Call by Reference.</p> <p>Function is good programming style in which we can write reusable code that can be called whenever require.</p> <p>Whenever we call a function then sequence of executable statements gets executed. We can pass some of the information to the function for processing called argument.</p>

Call by Value:

In case of call by value the function is invoked by a calling function, by passing the actual values to be processed.

This method copies the value of an argument into the formal parameters of the function.

By default, C programming language uses call by value method to pass arguments.

In general, this means that code within a function cannot alter the arguments used to call the function.

In this method, the changes made to parameter have no effect on the argument.

Program:

```
#include <stdio.h>
#include <conio.h>
int main() {
    int a, b, ans;
    int sum(int x, int y);
    printf("Enter Two Values:");
    scanf("%d%d", &a, &b);
    ans=sum(a,b);
    printf("sum=%d",ans);
    getch();
}
int sum (int x, int y) {
    int z;
    z = x+y;
    return(z);
}
```

Output:

Enter Two Values: 10 20

Sum = 30

Call by Reference:

In case of call by reference the function is invoked by a calling function, by passing the address of the actual values to be processed.

In this method, the address of an argument is copied in the parameter.

	<p>Inside the sub routine; the address is used to access the actual argument in the call.</p> <p>Program:</p> <pre> #include <stdio.h> #include <conio.h> int main() { int i, j; int swap(int*x, int*y); i=10; j=20; swap(&i, &j); printf("%d%d", i, j); getch(); } int swap (int*x, int*y) { int temp; temp = *y; *y = *x; *x = temp; } </pre> <p>Output:</p> <p>i=20 j=10</p>
38.	<p>Write a C program to find GCD of two numbers using recursion.</p> <p>Program:</p> <pre> #include <stdio.h> #include <conio.h> int hcf(int n1, int n2); int main() { int n1, n2; clrscr(); printf("Enter two positive integers: "); scanf("%d %d", &n1, &n2); printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2)); return 0; } </pre>

	<pre> int hcf(int n1, int n2) { if (n2 != 0) return hcf(n2, n1 % n2); else return n1; } </pre> <p>Run Code</p> <p>Output</p> <p>Enter two positive integers: 366 60</p> <p>G.C.D of 366 and 60 is 6.</p>
39.	<p>Write a C program to implement month name by accepting month number from user.(Use switch case)</p> <p>Program:</p> <pre> #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 is : January"); break; case 2: printf("Month is : February"); break; case 3: printf("Month is : March"); break; case 4: printf("Month is : April"); break; case 5: printf("Month is : May"); } } </pre>

	<pre> break; case 6: printf("Month is : June"); break; case 7: printf("Month is : July"); break; case 8: printf("Month is : August"); break; case 9: printf("Month is : September"); break; case 10: printf("Month is : October"); break; case 11: printf("Month is : November"); break; case 12: printf("Month is : December"); break; default : printf("invalid number"); } } </pre> <p>Output:</p> <p>Enter the month number : 9</p> <p>Month is : September</p>
40.	<p>Write a C program to accept 10 integers from the user and arrange them in ascending order and display them.</p> <p>Program:</p> <pre> #include<stdio.h> #include<conio.h> void main() { int i,j,temp,a[10]; clrscr(); printf("Enter 10 integer numbers: \n"); for(i=0;i<10;i++) scanf("%d",&a[i]); for (i=0;i<10;i++) </pre>

	<pre> { for(j=i+1;j<10;j++) { if(a[i]>a[j]) { temp=a[j]; a[j]=a[i]; a[i]=temp; } } } printf("\n\nThe 10 numbers sorted in ascending order are: \n"); for(i=0;i<10;i++) printf("%d\t",a[i]); getch(); } </pre> <p>OUTPUT:</p> <p>Enter 10 integer numbers:</p> <p>2 9 7 4 3 6 8 1 5 10</p> <p>The numbers 10 sorted in ascending order are:</p> <p>1 2 3 4 5 6 7 8 9 10</p>										
41.	<p>Give the difference between entry and exit controlled loop with an example.</p> <table border="1"> <thead> <tr> <th>Entry Controlled Loop</th><th>Exit Controlled Loop</th></tr> </thead> <tbody> <tr> <td>Test condition is checked first, and then loop body will be executed.</td><td>Loop body will be executed first, and then condition is checked.</td></tr> <tr> <td>If Test condition is false, loop body will not be executed.</td><td>If Test condition is false, loop body will be executed once.</td></tr> <tr> <td>for loop and while loop are the examples of Entry Controlled Loop.</td><td>do while loop is the example of Exit controlled loop.</td></tr> <tr> <td>Entry Controlled Loops are used when checking of test condition is mandatory before executing loop body.</td><td>Exit Controlled Loop is used when checking of test condition is mandatory after executing the loop body.</td></tr> </tbody> </table> <p>Using while loop</p> <pre>int count=100;</pre>	Entry Controlled Loop	Exit Controlled Loop	Test condition is checked first, and then loop body will be executed.	Loop body will be executed first, and then condition is checked.	If Test condition is false, loop body will not be executed.	If Test condition is false, loop body will be executed once.	for loop and while loop are the examples of Entry Controlled Loop.	do while loop is the example of Exit controlled loop.	Entry Controlled Loops are used when checking of test condition is mandatory before executing loop body.	Exit Controlled Loop is used when checking of test condition is mandatory after executing the loop body.
Entry Controlled Loop	Exit Controlled Loop										
Test condition is checked first, and then loop body will be executed.	Loop body will be executed first, and then condition is checked.										
If Test condition is false, loop body will not be executed.	If Test condition is false, loop body will be executed once.										
for loop and while loop are the examples of Entry Controlled Loop.	do while loop is the example of Exit controlled loop.										
Entry Controlled Loops are used when checking of test condition is mandatory before executing loop body.	Exit Controlled Loop is used when checking of test condition is mandatory after executing the loop body.										

```
while(count<50)
printf("%d",count++);
```

Using for loop

```
int count;
for(count=100; count<50; count++)
printf("%d",count);
```

In both code snippets value of count is 100 and condition is count<50, which will check first, hence there is no output.

Using do while loop

```
int count=100;
do
{
printf("%d",count++);
}while(count<50);
```

In this code snippet value of count is 100 and test condition is count<50 which is false yet loop body will be executed first then condition will be checked after that. Hence output of this program will be 100.