CHANDIGARH ENGINEERING COLLEGE

Department of Applied Sciences

PPS LAB MANUAL

Lab Name – Programming for Problem Solving Lab
Subject Code – BTPS-102-18

Common to All Branches
ECE/ME/CSE/IT/AI-DS/ AI-ML

Branch – ECE/IT/AI-ML/AI-DS
Semester – 1st

Branch – CSE/ME
Semester – 2nd

Lab Instructor Lab Incharge

Verified by:
Head of Department

PPS Lab – BTPS-102-18

Syllabus As per I.K.Gujral Punjab Technical University

1. Familiarization with programming environment.
2. Write a Program to print text on the screen.
3. Write a Program to perform arithmetic operation
4. Write a Program to swap two numbers with the help of third variable and without using of third variable
5. Write a Program to find the entered number is even or odd
6. Write a Program to find the largest among three numbers.
7. Write a Program to find the roots of quadratic equation.
8. Write a Program to find whether the entered character is upper case , lower case , digit or special symbol.
9. Write a Program to find the sum of a series 1n+2n+3n+……+mn
10. Write a Program to find factorial of given number by using while loop.
11. Write a Program to check whether the entered number is palindrome or not by using do-while.
12. Write a Program to perform arithmetic operation on 1-D Array.
13. Write a Program to perform arithmetic operation on 2-D Array.
14. Write a Program to perform string manipulation function.
15. Write a Program to search an element using linear or binary search.
16. Write a Program to sort list of elements using Bubble sort.
17. Write a Program to sort list of elements using Insertion sort.
18. Write a Program to sort list of elements using Selection sort.
19. Write a Program to swap two numbers using call by value and call by reference.
20. Write a Program to find Fibonacci series using recursion.
21. Write a Program to sort list of elements using Quick sort.
22. Write a Program to sort list of elements using Merge sort.
23. Write a Program to perform Array of Structure.
24. Write a Program to implement pointers.
25. Write a Program to perform file handling.

List of Experiments with objective details with Real Time Applications

1. Familiarization with programming environment.
   To get introduction to the programming environment. To learn the basics of programming language.

2. Write a Program to print text on the screen.
   To understand the syntax of displaying text in C language.

3. Write a Program to perform arithmetic operation.
   To understand the execution of basic arithmetic operations as used in all software such as calculator programs and excel commands.

4. Write a Program to swap two numbers with the help of third variable and without using of third variable
   To understand the process of exchanging values with two variables as used in copy paste commands and move commands.

5. Write a Program to find the entered number is even or odd.
   To understand how to apply mathematical calculations using simple if statement (Conditional commands).

6. Write a Program to find the largest among three numbers.
   To understand how to apply mathematical calculations using different conditional statements.

7. Write a Program to find the roots of quadratic equation.
   To understand how to apply mathematical calculations using different if statements.

8. Write a Program to find whether the entered character is upper case, lower case, digit or special symbol.
   To understand the usage of keys in keyboard and how they are processed in international standard codes.

9. Write a Program to find the sum of a series $1^n+2^n+3^n+\ldots\ldots+m^n$
   To understand how to apply mathematical calculations using simple loop statement.

10. Write a Program to find factorial of given number by using while loop.
    To understand the usage of simple while loop to repeat the execution of some statements.

11. Write a Program to check whether the entered number is palindrome or not by using do-while.
    To understand the usage of simple do while loop to repeat the execution of some statements.

12. Write a Program to perform arithmetic operation on 1-D Array.
    To understand how to store details in contiguous locations and usage of array.

13. Write a Program to perform arithmetic operation on 2-D Array.
    To understand how to use arrays for functions such as matrix operations.

14. Write a Program to perform string manipulation function.
    To understand how to use strings and use string functions such as used in password visibility as special symbols in any e-authentications, ignorance of uppercase and lowercase in emailIDS .

15. Write a Program to search an element using linear or binary search.
    To understand how to search a particular detail from the given list of information such as used in CtrlF command and the process of search engines.

16. Write a Program to sort list of elements using Bubble sort.
    To understand how to arrange the list of records using basic sorting algorithms. The similar way the records are listed by the search engine, the commands used to arrange the data in an excel sheet.

17. Write a Program to sort list of elements using Insertion sort.
    To understand how to arrange the list of records using basic sorting algorithms. The similar way the records are listed by the search engine, the commands used to arrange the data in an excel sheet.

18. Write a Program to sort list of elements using Selection sort.
    To understand how to arrange the list of records using basic sorting algorithms. The similar way the records are listed by the search engine, the commands used to arrange the data in an excel sheet.

19. Write a Program to swap two numbers using call by value and call by reference.
    To understand the process of exchanging values of two variables and differences in calling a function by passing the value of the variables and passing address of the variable.

20. Write a Program to find Fibonacci series using recursion.
    To understand how to repeat the execution of a function by calling the function itself and applying in finding basic mathematical series.

21. Write a Program to sort list of elements using Quick sort.
    To understand how to arrange the list of records using basic sorting algorithms using recursive process. The similar way the records are listed by the search engine, the commands used to arrange the data in an excel sheet.

22. Write a Program to sort list of elements using Merge sort.

To understand how to arrange the list of records using basic sorting algorithms using recursive process. The similar way the records are listed by the search engine, the commands used to arrange the data in an excel sheet.

23. Write a Program to perform Array of Structure.
    To understand syntax of structures in C and grouping variables of different datatype together.

24. Write a Program to implement pointers.
    To understand the usage of pointer variables and referring a variable using address.

25. Write a Program to perform file handling.
    To understand how to create a file and how to use file functions.

Practical Beyond Syllabus

Discussion on Introduction to C++

- Conversion of C program into C++ program
- Introduction to Code Block IDE
- Introduction to Dev C IDE

Extra Programs in C

1. Find the largest of 2 no. using condition operator
2. To demonstrate the importance of ampersand(&)operator in scanf
3. To demonstrate the work of %n
4. Display decimal, octal & hexadecimal no.
5. Check the no. is positive or negative
6. Given character is digit or not
7. Check the no.is divided by 5 or 7 or by both or by no one
8. To demonstrate the working of preprocessor
9. To demonstrate the use of bitfields
10. Demonstrate the use of mallock, callock and free function
11. To explain the use of static storage classes
12. To show different between a and &a
13. Check the value is vowel or consonant
14. Read N no. from user and find average
15. Factorial of given no.
16. Convert binary no. to decimal no.
17. Convert decmal no. to binary no.
18. Reverse of no.
19. Check the no. is amstrong or not
20. To find the sum of series 1+ ${}^{1}/_{2} + {}^{1}/_{3} + {}^{1}/_{4} + \ldots\ldots {}^{1}/_{n}$
21. To find the sum of series 1+x $+ \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \cdots + \frac{x^n}{n!}$

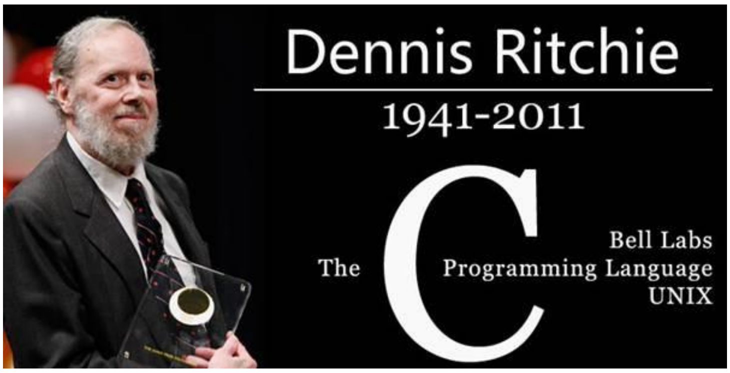
List of Extra-Experiments with objective details with Real Time Applications

1. Find the largest of 2 no. using condition operator
   To understand the usage of conditional operator in C.
2. To demonstrate the importance of ampersand(&)operator in scanf
   To understand the usage of & in reading input value and storing the value in the respective address.
3. To demonstrate the work of %n
   To understand the option to count number of characters in a string without built-in function.
4. Display decimal, octal & hexadecimal no.
   To understand number system and displaying the numbers in different format.
5. Check the no. is positive or negative
   To practice basic concepts of if else statements in checking sign of numbers
6. Given character is digit or not
   To understand type of character
7. Check the no.is divided by 5 or 7 or by both or by no one
   To practice else if statements in C.
8. To demonstrate the working of preprocessor
   To understand the use of preprocessor in C
9. To demonstrate the use of bitfields
   To understand the usage of memory allocation for structure variable and usage of bit fields
10. Demonstrate the use of mallock, callock and free function
    To understand the usage of pointer function in memory allocation
11. To explain the use of static storage classes
    To understand the usage of storage classes in C.
12. To show different between a and &a
    To understand the difference between storing a variable value using & and without &
13. Check the value is vowel or consonant
    To practice switch cases in C
14. Read N no. from user and find average
    To practice mathematical calculations using arrays
15. Factorial of given no.
    To practice mathematical operations using different loops, functions and a recursive function
16. Convert binary no. to decimal no.
    To practice number system and conversion
17. Convert decmal no. to binary no.
    To practice number system and conversion in either way

18. Reverse of no.
    To apply mathematical functions in displaying a number(basic functions used in excel and so on)
19. Check the no. is amstrong or not
    To practice mathematical calculations of numbers.
20. To find the sum of series 1+ $^1/_2 + ^1/_3 + ^1/_4 + \ldots\ldots\ ^1/_n$
    To practice mathematical operations using loop
21. To find the sum of series 1+x $+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\cdots+\frac{x^n}{n!}$
    To understand built-in mathematical functions in header file math.h

Practical 1: Familiarization with programming environment

Aim: To get Familiarization with programming environment



AT&T
Bell Laboratories

C is a general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion. By design, C provides constructs that map efficiently to typical machine instructions, and therefore it has found lasting use in applications that had formerly been coded in assembly language, including operating systems, as well as various application software for computers ranging from supercomputers to embedded systems.

C was originally developed by Dennis Ritchie between 1969 and 1973 at Bell Labs, and used to re-implement the Unix operating system. It has since become one of the most widely used programming languages of all time, with C compilers from various vendors available for the majority of existing computer architectures and operating systems. C has been standardized by the American National Standards Institute (ANSI) since 1989 (see ANSI C) and subsequently by the International Organization for Standardization (ISO).

C is an imperative procedural language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run-time support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant and portably written C program can be compiled for a very wide variety of computer platforms and operating systems with few changes to its source code. The language has become available on a very wide range of platforms, from embedded microcontrollers to supercomputers.

IDE: An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of a **source code editor**, build automation tools, and a debugger.

Compiler: A compiler is a software program that compiles program source code files into an executable program. It is included as part of the integrated development environment IDE with most programming software packages.

The compiler takes source code files that are written in a high-level language, such as C, BASIC, or Java, and compiles the code into a low-level language, such as machine code or assembly code. This code is created for a specific processor type, such as and Intel
Pentium or PowerPC. The program can then be recognized by the processor and run from the operating system.

C++ built-in Header Files				
alloc.h	dos.h	limits.h	share.h	sys\stat.h
assert.h	errno.h	locale.h	signal.h	sys\timeb.h
bcd.h	fcntl.h	malloc.h	stdarg.h	sys\types.h
bios.h	float.h	math.h	stddef.h	time.h
complex.h	fstream.h	mem.h	stdio.h	utime.h
conio.h	generic.h	memory.h	stdiostr.h	values.h
ctype.h	graphics.h	new.h	stream.h	varargs.h
dir.h	io.h	process.h	string.h	sys\stat.h
direct.h	iomanip.h	search.h	strstrea.h	sys\timeb.h
dirent.h	iostream.h	setjmp.h	sys\locking.h	sys\types.h

List of best and free C/C++ compilers and IDEs for Programmers			
Eclipse	Dev C++	GCC	CC65
NetBeans	CodeLite	Xcode	LCC
Code::Blocks	MinGW	Tiny C – Compiler	SDCC
Digital Mars:	Ultimate++	Portable C Compiler	nesC
C Free	Mirosoft Visual Studio Express	Failsafe C	CC386
Sky IDE	Open Watcom	Pelles C	SubC
Turbo C++	TC	TurboC3	

Turbo C++ Keyboard Shortcuts					
Shortcuts keys	**Action**	**Shortcuts keys**	**Action**	**Shortcuts keys**	**Action**
F1	For Help	Alt+X	Quit	Ctrl+F9	Run
F2	Save	Alt+Bksp	Undo	Ctrl+F2	Program reset
F3	Open	Shift+Alt+Bksp	Redo	Alt+F9	Compile

F4	Go to cursor	Shift+Del	Cut	Alt+F4	Inspect
F5	Zoom	Ctrl+Ins	Copy	Ctrl+F4	Evaluate/Modify
F6	Next	Shift+Ins	Paste	Ctrl+F3	Call stack
F7	Trace into	Ctrl+Del	Clear	Ctrl+F8	Toggle breakpoint
F8	Step over	Ctrl+L	Search again	Ctrl+F5	Size/Move
F9	Make	Alt+F7	Previous error	Alt+F3	Close
F10	Menu	Alt+F8	Next error	Alt+F5	User screen
Alt+0	List all	Shift+F1	Index	Ctrl+F7	Add watchb
Ctrl+F1	Topic search	Alt+F1	Previous topic		

Data Types available in C++:

1. Primary(Built-in) Data Types:
   a. character
   b. integer
   c. floating point
   d. boolean
   e. double floating point
   f. void
   g. wide character
2. User Defined Data Types:
   a. Structure
   b. Union
   c. Class
   d. Enumeration
3. Derived Data Types:
   a. Array
   b. Function
   c. Pointer
   d. Reference

The lists of modifiers used in C++ are:

signed
unsigned
long
short

Data Type (Keywords)	Description	Size	Typical Range
Char	Any single character. It may include a letter, a digit, a punctuation mark, or a space.	1 byte	-128 to 127 or 0 to 255
signed char	Signed character.	1 byte	-128 to 127
unsigned char	Unsigned character.	1 byte	0 to 255
wchar_t	Wide character.	2 or 4 bytes	1 wide character
Int	Integer.	4 bytes	-2147483648 to 2147483647
signed int	Signed integer. Values may be negative, positive, or zero.	4 bytes	-2147483648 to 2147483647
unsigned int	Unsigned integer. Values are always positive or zero. Never negative.	4 bytes	0 to 4294967295
long	Long integer.	4 bytes	-2147483648 to 2147483647
signed long	Signed long integer. Values may be negative, positive, or zero.	4 bytes	-2147483648 to 2147483647
unsigned long	Unsigned long integer. Values are always positive or zero. Never negative.	4 bytes	0 to 4294967295

Floating-point Data Types

Data Type (Keywords)	Description	Size	Typical Range
float	Floating point number. There is no fixed number of digits before or after the decimal point.	4 bytes	+/- 3.4e +/- 38 (~7 digits)
double	Double precision floating point number. More accurate compared to float.	8 bytes	+/- 1.7e +/- 308 (~15 digits)
long double	Long double precision floating point number.	8 bytes	+/- 1.7e +/- 308 (~15 digits)

Boolean Data Type

Data Type (Keywords)	Description	Size	Typical Range
bool	Boolean value. It can only take one of two values: true or false.	1 byte	true or false

C Keywords			
asm	else	New	This
auto	enum	Operator	Throw
bool	explicit	Private	TRUE
break	export	Protected	Try
case	extern	Public	Typedef
catch	FALSE	Register	Typeid
char	float	reinterpret_cast	Typename
class	for	Return	Union
const	friend	Short	Unsigned
const_cast	goto	Signed	Using
continue	if	Sizeof	Virtual
default	inline	Static	Void
delete	int	static_cast	Volatile

do	long	Struct	wchar_t
double	mutable	Switch	While
dynamic_cast	namespace	Template	
In addition, the following words are also reserved:			
And	bitor	not_eq	Xor
and_eq	compl	Or	xor_eq
bitand	not	or_eq	

Functions	Descriptions
abort	stops the program
abs	absolute value
acos	arc cosine
asctime	a textual version of the time
asin	arc sine
assert	stops the program if an expression isn't true
atan	arc tangent
atan2	arc tangent, using signs to determine quadrants
atexit	sets a function to be called when the program exits
atof	converts a string to a double
atoi	converts a string to an integer
atol	converts a string to a long
bsearch	perform a binary search
calloc	allocates and clears a two-dimensional chunk of memory
ceil	the smallest integer not less than a certain value
clearerr	clears errors
clock	returns the amount of time that the program has been running
cos	Cosine
cosh	hyperbolic cosine
ctime	returns a specifically formatted version of the time
difftime	the difference between two times
div	returns the quotient and remainder of a division
exit	stop the program
exp	returns "e" raised to a given power
fabs	absolute value for floating-point numbers
fclose	close a file

feof	true if at the end-of-file
ferror	checks for a file error
fflush	writes the contents of the output buffer
fgetc	get a character from a stream
fgetpos	get the file position indicator
fgets	get a string of characters from a stream
floor	returns the largest integer not greater than a given value
fmod	returns the remainder of a division
fopen	open a file
fprintf	print formatted output to a file
fputc	write a character to a file
fputs	write a string to a file
fread	read from a file
free	returns previously allocated memory to the operating system
freopen	open an existing stream with a different name
frexp	decomposes a number into scientific notation
fscanf	read formatted input from a file
fseek	move to a specific location in a file
fsetpos	move to a specific location in a file
ftell	returns the current file position indicator
fwrite	write to a file
getc	read a character from a file
getchar	read a character from STDIN
getenv	get environment information about a variable
gets	read a string from STDIN
gmtime	returns a pointer to the current Greenwich Mean Time
isalnum	true if a character is alphanumeric
isalpha	true if a character is alphabetic
iscntrl	true if a character is a control character
isdigit	true if a character is a digit
isgraph	true if a character is a graphical character
islower	true if a character is lowercase
isprint	true if a character is a printing character
ispunct	true if a character is punctuation
isspace	true if a character is a space character
isupper	true if a character is an uppercase character
itoa	Convert a integer to a string
isxdigit	true if a character is a hexadecimal character
labs	absolute value for long integers
ldexp	computes a number in scientific notation
ldiv	returns the quotient and remainder of a division, in long integer form

localtime	returns a pointer to the current time
log	natural logarithm
log10	natural logarithm, in base 10
longjmp	start execution at a certain point in the program
malloc	allocates memory
memchr	searches an array for the first occurrence of a character
memcmp	compares two buffers
memcpy	copies one buffer to another
memmove	moves one buffer to another
memset	fills a buffer with a character
mktime	returns the calendar version of a given time
modf	decomposes a number into integer and fractional parts
perror	displays a string version of the current error to STDERR
pow	returns a given number raised to another number
printf	write formatted output to STDOUT
putc	write a character to a stream
putchar	write a character to STDOUT
puts	write a string to STDOUT
qsort	perform a quicksort
raise	send a signal to the program
rand	returns a pseudo-random number
realloc	changes the size of previously allocated memory
remove	erase a file
rename	rename a file
rewind	move the file position indicator to the beginning of a file
scanf	read formatted input from STDIN
setbuf	set the buffer for a specific stream
setjmp	set execution to start at a certain point
setlocale	sets the current locale
setvbuf	set the buffer and size for a specific stream
signal	register a function as a signal handler
sin	Sine
sinh	hyperbolic sine
sprintf	write formatted output to a buffer
sqrt	square root
srand	initialize the random number generator
sscanf	read formatted input from a buffer
strcat	concatenates two strings
strchr	finds the first occurrence of a character in a string
strcmp	compares two strings

strcoll	compares two strings in accordance to the current locale
strcpy	copies one string to another
strcspn	searches one string for any characters in another
strerror	returns a text version of a given error code
strftime	returns individual elements of the date and time
strlen	returns the length of a given string
strncat	concatenates a certain amount of characters of two strings
strncmp	compares a certain amount of characters of two strings
strncpy	copies a certain amount of characters from one string to another
strpbrk	finds the first location of any character in one string, in another string
strrchr	finds the last occurrence of a character in a string
strspn	returns the length of a substring of characters of a string
strstr	finds the first occurrence of a substring of characters
strtod	converts a string to a double
strtok	finds the next token in a string
strtol	converts a string to a long
strtoul	converts a string to an unsigned long
strxfrm	converts a substring so that it can be used by string comparison functions
system	perform a system call
tan	Tangent
tanh	hyperbolic tangent
time	returns the current calendar time of the system
tmpfile	return a pointer to a temporary file
tmpnam	return a unique filename
tolower	converts a character to lowercase
toupper	converts a character to uppercase
ungetc	puts a character back into a stream
va_arg	use variable length parameter lists
vprintf, vfprintf, and vsprintf	write formatted output with variable argument lists
vscanf, vfscanf, and vsscanf	read formatted input with variable argument lists

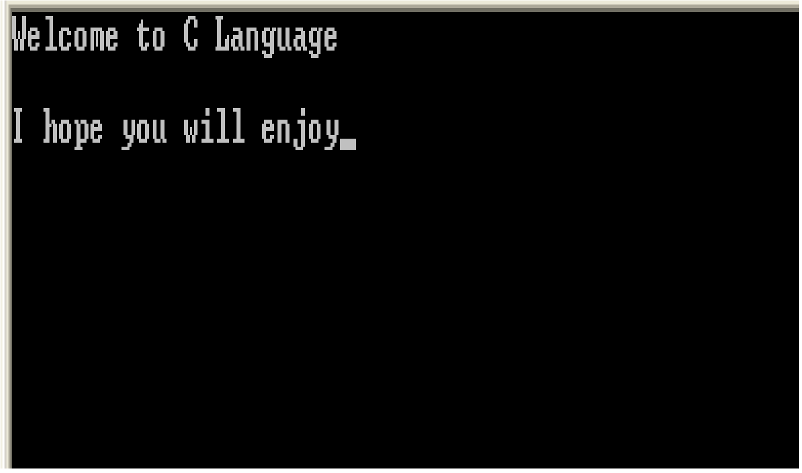
Practical 2: WAP to print text on the screen.

Aim: To write a C program to print text on the screen

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("Welcome to C Language\n\n");
printf("I hope you will enjoy");
getch();
}
```

Output:



```
Welcome to C Language

I hope you will enjoy_
```

Result: The C program to print text on the screen was executed and output was verified.

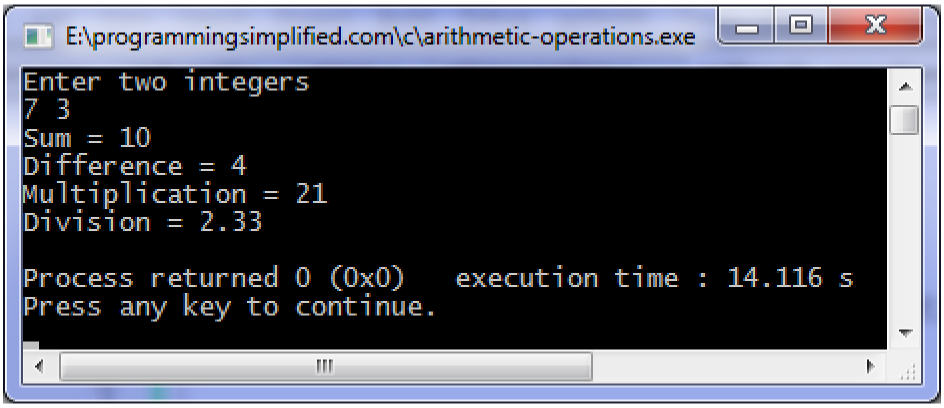
Practical 3: Write a C program to perform arithmetic operation

Aim: To write a C program to perform arithmetic operations

Program:

```
#include <stdio.h>
int main()
{
int first, second, add, subtract, multiply;
float divide;

printf("Enter two integers\n");
scanf("%d%d", &first, &second);

add = first + second;
subtract = first - second;
multiply = first * second;
divide = first / (float)second;   //typecasting

printf("Sum = %d\n", add);
printf("Difference = %d\n", subtract);
printf("Multiplication = %d\n", multiply);
printf("Division = %.2f\n", divide);

return 0;
}
```

Output :



```
Enter two integers
7 3
Sum = 10
Difference = 4
Multiplication = 21
Division = 2.33

Process returned 0 (0x0)   execution time : 14.116 s
Press any key to continue.
```

Result: The C program to perform arithmetic operations were executed and output was verified

Practical 4: Write a C program to swap two numbers with the help of third variable and without using of third variable

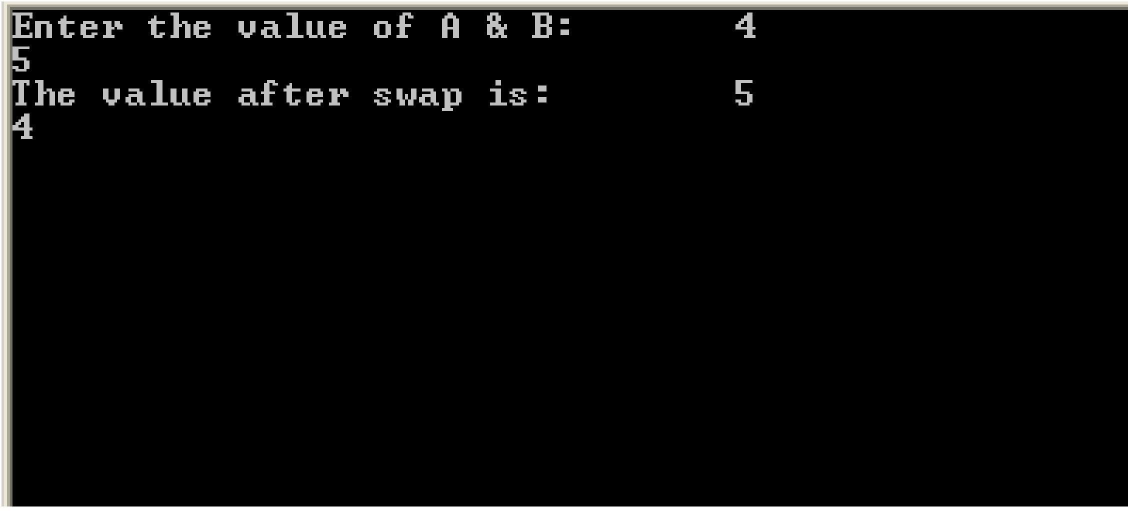
4.1 Aim: To write a C program to swap two numbers with the help of third variable.

Program:

```
#include<stdio.h>//standard input output
#include<conio.h>//console input output
int main( )
{
int a,b,c;
clrscr( );//clear screen
printf("Enter the value of A & B:\t");
scanf("%d%d",&a,&b);
 c=a;
 a=b;
 b=c;
printf("The value after swap is:\t%d\n%d",a,b);
getch( );//to get character from the user as input
return 0;
}
```

a	b	Statement

Output



```
Enter the value of A & B:        4
5
The value after swap is:         5
4
```

Result: The C program to swap two numbers with the help of third variable was executed and output was verified.

4.2 Aim: To write a C program to swap two numbers without using a third variable

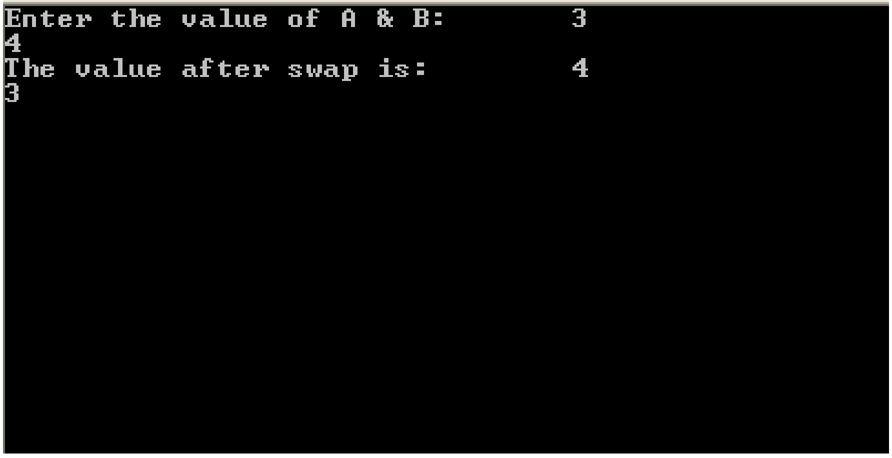
Program:

```
#include<stdio.h>
#include<conio.h>
void main()
```

```
{
int a,b;
clrscr();
printf("Enter the value of A & B:\t");
scanf("%d%d",&a,&b);
 a=a+b;
 b=a-b;
 a=a-b;
printf("The value after swap is:\t%d\n%d",a,b);
getch();
}
```

3	4	scanf
7	4	a=a+b
7	3	b=a-b
4	3	a=a-b

Output



```
Enter the value of A & B:       3
4
The value after swap is:        4
3
```

Result: The C program to swap two numbers without using a third variable was executed and output was verified.

Practical 5: Write a C program to find whether the entered number is even or odd

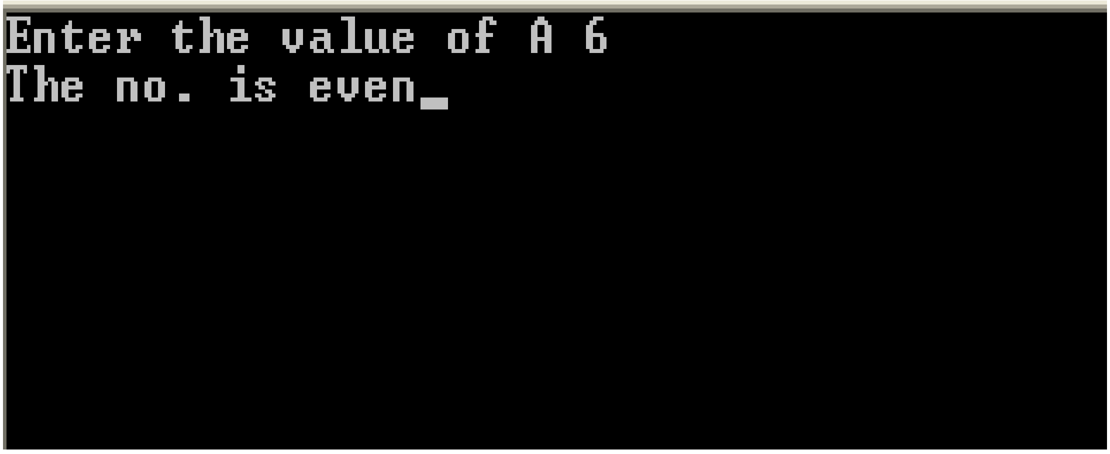
Aim: To write a C program to find whether the entered number is even or odd

Program:

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

```
#include<conio.h>
void main()
{
int a;
clrscr();
printf("Enter the value of A ");
scanf("%d",&a);
if(a%2==0)
printf("The no. is even");
else
printf("the no. is odd");
getch();
}
```

Output



```
Enter the value of A 6
The no. is even_
```

Result: The C program to find the whether the entered number is even or odd was executed and output was verified

Practical 6: Write a C program to find the largest among three numbers

Aim: To write a C program to find the largest among three numbers

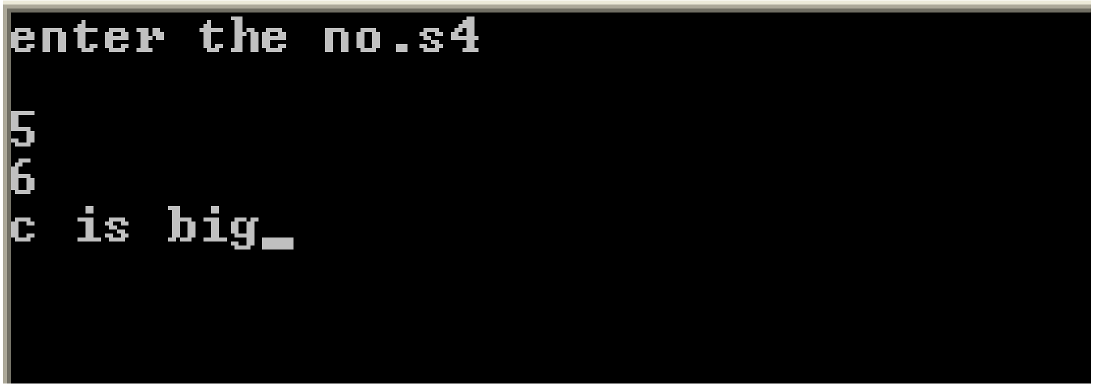
Program:
Method1:

```
#include<stdio.h>
int main()
{
int a,b,c;
printf("Enter three different no.s\n");
scanf("%d%d%d",&a,&b,&c);
if(a>b)
 {
if(a>c)
printf("%d is greatest among the numbers %d ,%d  and %d",a,a,b,c);
else
printf("%d is big",c);
 }
else
{
if(b>c)
printf("%d is big",b);
else
printf("%d  is big",c);
}
return 0;
}
```

Method2:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c,max;
clrscr();
printf("enter the no.s");
scanf("%d%d%d",&a,&b,&c);
if(a>b)
max=a;
else
max=b;

if(max>c)
printf("%d is greatest",max);
else
printf("%d is greatest",c);
getch();
}
```

Output



```
enter the no.s4

5
6
c is big_
```

Result: The C program to find the find the largest among three numbers was executed and output was verified

Practical 7: Write a C program to find the roots of quadratic equation.

Aim: To write a C program to find the roots of quadratic equation

Quadratic Equation:

$ax^2+bx+c=0$

Roots:
[-b+-sqrt((b*b)-(4*a*c))]/(2*a)

1. Root1→[-b+sqrt((b*b)-(4*a*c))]/(2*a)
2. Root2→[-b-sqrt((b*b)-(4*a*c))]/(2*a)

Input: Coefficients of the equation
a,b and c

Processing:
Calculate- b*b-(4*a*c) equate to d
Case1:d<0
Roots are imaginary

Case2: d==0
Roots are equal

Caes3: d>0
Different roots, R1 and R2

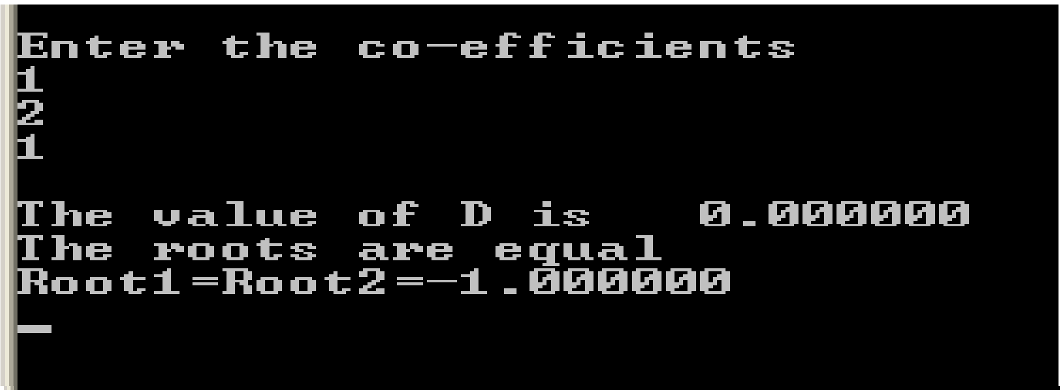
Output: Roots R1 and R2

Program:

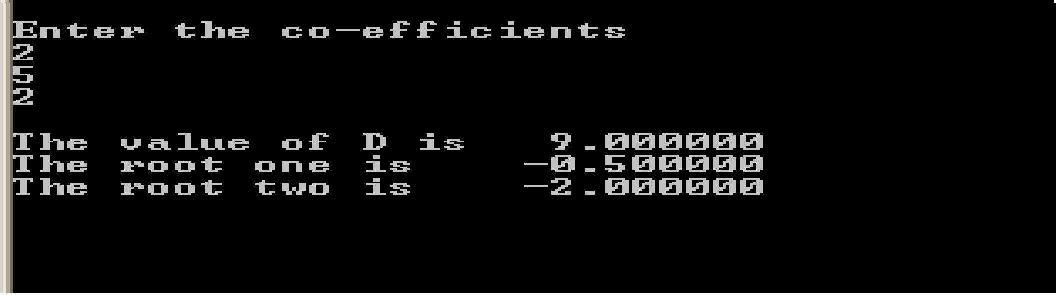
```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
float a,b,c,r,r1,r2,d;
clrscr();
printf("\nEnter the co-efficients\n");
scanf("%f%f%f",&a,&b,&c);
d=(b*b)-(4*a*c);
printf("\nThe value of d is  %f",d);
if(d<0)
printf("\nThe root are not real");
if(d>0)
{r1=(-b+sqrt(d))/(2*a);
r2=(-b-sqrt(d))/(2*a);
printf("\nThe root one is    %f",r1);
printf("\nThe root two is    %f",r2);
}
if(d==0)
{
```

```
r=-b/(2*a);
printf("\nThe roots are equal\n");
printf("Root1=Root2=%f\n",r);
}
getch();
}
```

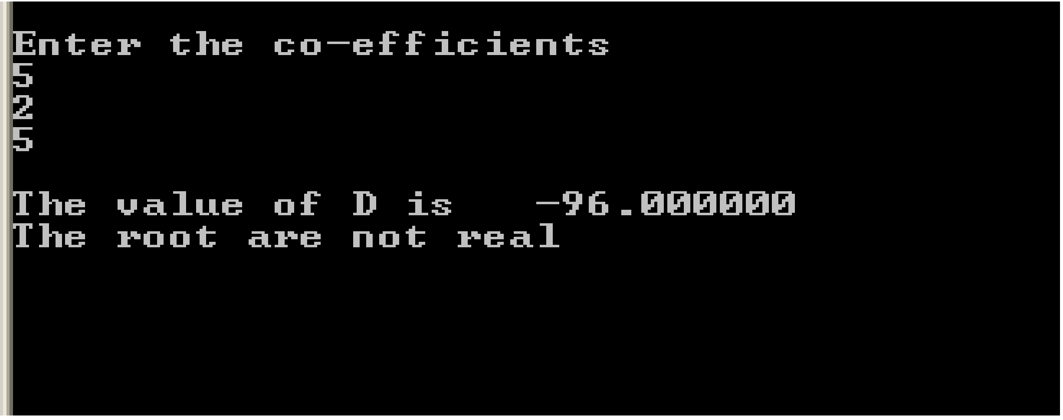
Output



```
Enter the co-efficients
1
2
1

The value of D is    0.000000
The roots are equal
Root1=Root2=-1.000000
_
```



```
Enter the co-efficients
2
5
2

The value of D is    9.000000
The root one is     -0.500000
The root two is     -2.000000
```



```
Enter the co-efficients
5
2
5

The value of D is    -96.000000
The root are not real
```

Result: The C program to find the find the roots of a quadratic equation was executed and output was verified

Practical 8: Write a C program to find whether the entered character is upper case , lower case , digit or special symbol.

Aim: To write a C program to find whether the entered character is upper case , lower case , digit or special symbol.

Characters	ASCII VAlues
A-Z	65-90
a-z	97-122
0-9	48-57
Special symbols	0-47, 58-64, 91-96, 123-127

```
#include<stdio.h>
int main()
{
char ch;
printf("Enter a character:");
scanf("%c",&ch);
if(ch>=65 &&ch<=90)
{
printf("\n Upper case letter");
}
else if(ch>=97 &&ch<=122)
{
printf("\n Lower case letter");
}
else if(ch>=48 &&ch<=57)
{
printf("\n Digit");
}
else if((ch>=0 &&ch<=47) || (ch>=58&&ch<=64) || (ch>=91 &&ch<=96) || (ch>=123
&&ch<=127))
{
printf("\n Special symbol");
}
else
printf(“Invalid”);
return 0;
}
```

Output1:

```
Enter a Character: H
Upper case letter
```

Output2:

```
Enter a Character: g
Lower case letter
```

Output3:

```
Enter a Character: 8
Digit
```

Output4:

```
Enter a Character: @
Special symbol
```

Result: The C program to find the whether the entered character is upper case , lower case , digit or special symbol was executed and output was verified

Practical 9: Write a C program to find the sum of given series $1^n+2^n+3^n+\ldots\ldots+m^n$

Aim: To write a C program to find the sum of given series $1^n+2^n+3^n+\ldots\ldots+m^n$

Given Series-----→**$1^n+2^n+3^n+\ldots\ldots+m^n$**

Input- m, n
Output -Sum- of series
Looping Statement- for, while, do while (Any of these loops can be used)
*****Write any one program in your file. Practice all three methods.**

1. Using For looping statement:

Program:

Sample Value and Iteration step

```
#include<stdio.h>
#include<math.h>
int main()
{
int  m,n,sum=0,i,term;
printf("Enter the value of m and n");
scanf("%d%d",&m,&n);
for( i=1;i<=m ;i++)
  {
term=pow(i,n);   //i^n;
sum=sum+term;  //sum=sum+pow(i,n);
    }
printf("The sum of given series is %d",sum);
return 0;
}
```

2. Using while looping statement:

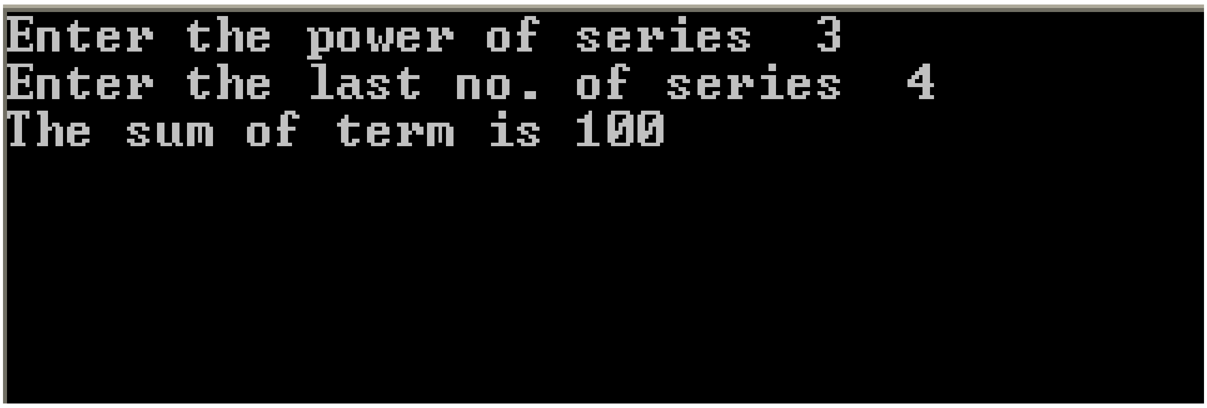
```
#include<stdio.h>
#include<math.h>
int main()
{
int n,m,sum=0,i=1,term;
printf("Enter the power of series  ");
scanf("%d",&n);
printf("Enter the last no. of series  ");
scanf("%d",&m);
while(i<=m)
 {
term=pow(i,n);
sum=sum+term;
i++;
 }
printf("The sum of term is %d",sum);
return 0;
}
```

3. Using do - while looping statement:

```
#include<stdio.h>
#include<math.h>
int main()
```

```
{
int n,m,sum=0,i=1,term;
printf("Enter the power of series  ");
scanf("%d",&n);
printf("Enter the last no. of series  ");
scanf("%d",&m);
do
 {
term=pow(i,n);
sum=sum+term;
i++;
 } while(i<=m);
printf("The sum of term is %d",sum);
return 0;
}
```

Output



```
Enter the power of series  3
Enter the last no. of series  4
The sum of term is 100
```

Result: The C program to find the sum of given series $1^n+2^n+3^n+\ldots\ldots+\ m^n$ was executed and output was verified

Practical 10: Write a C program to find the factorial of given number by using while loop

Aim: To write a C program to find factorial of given number by using while loop.

Input- a number n

Output –Factorial of n
Looping Statement- while

Program:

```
#include<stdio.h>
#include <conio.h>
void main()
{
int n,f=1,i=1;
printf("Enter a number: ");
scanf("%d",&n);
while(i<=n)
    {
        f = f * i;
i++;
    }
printf("\n Factorial of %d = %d", n,f);
getch();

}
```

Output:

```
Enter a number: 3
Factorial of 3 = 6
```

Result: The C program to find factorial of given number by using while loop was executed and output was verified

Practical 11: Write a C program to check whether the entered number is palindrome or not by using do-while.

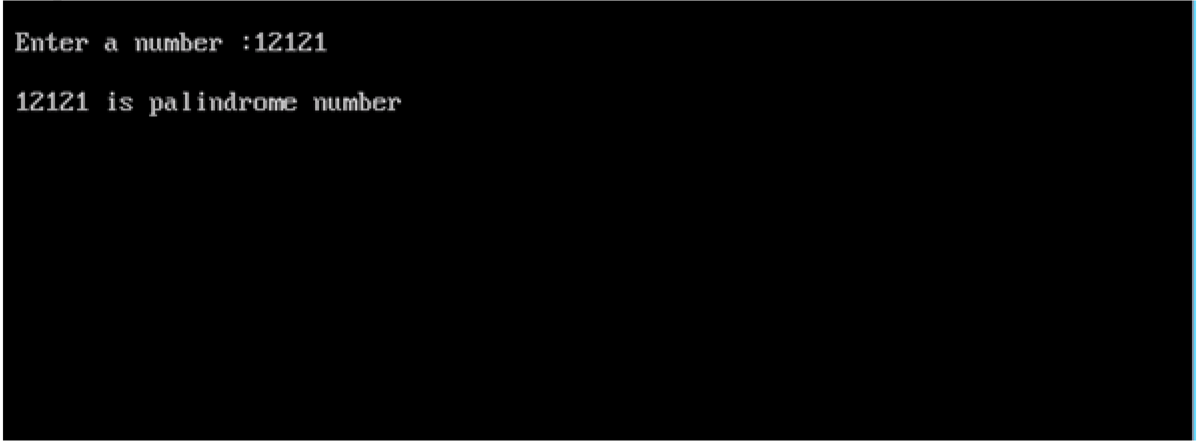
Aim: To write a C program to check whether the entered number is palindrome or not by using do-while.

Input: Number
Process:1. Reverse of Input Number
2.Check if input Number is equal to the Reversed Number to find if Palindrome or Not
Output: Display the result

Program:

```
#include<stdio.h>
#include<conio.h>
void main()
{
  int num,rev=0,rem,original;
  printf("\n Enter a number: ");
  scanf("%d",&num);
original=num;
do
  {
    rem=num%10;
    rev=rev*10+rem;
    num=num/10;
  }
while(num>0);
if(rev==original)
printf("The given number %d is a Palindrome",original);
else
printf("The given number %d is not a Palindrome",original);
getch();
}
```

Output:



```
Enter a number :12121

12121 is palindrome number
```

Result: The C program to check whether the entered number is palindrome or not by using do-while was executed and output was verified

Practical 12: Write a C program to perform arithmetic operation on 1-D Array.

Aim: To write a C program to perform arithmetic operation on 1-D Array

Program:

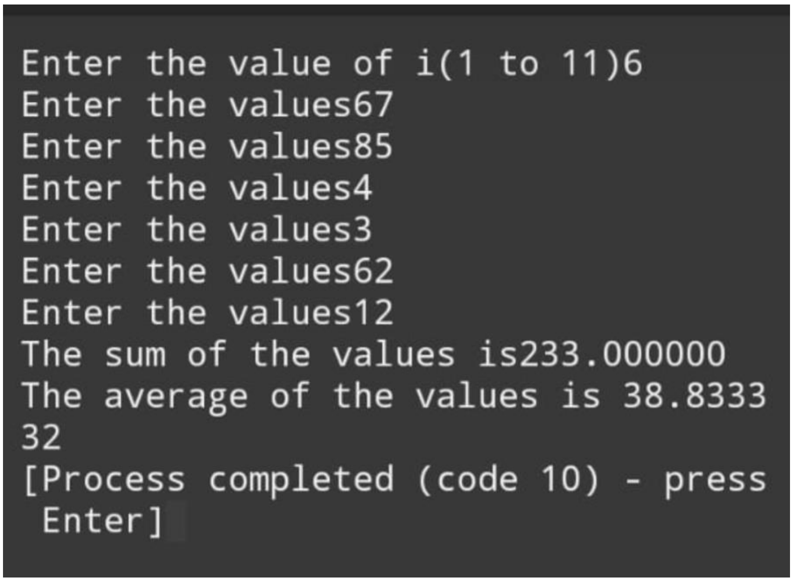
```
#include<stdio.h>
```

```
#include<conio.h>
void main()
{
int a[11],b,i;

float sum=0,avg;
clrscr();
printf("Enter the value of i(1 to 11) ");
scanf("%d",&i);

for(b=0;b<i;b++)
{ printf("Enter the values ");
scanf("%d",&a[b]);
sum=sum+a[b];
}
printf("The sum of the values is %f\n",sum);
avg=sum/i;
printf("The average of the values is %f",avg);
getch();
}
```

a[0]	a[1]	a[2]	a[3]	a[4]
4	6	7	8	9

i	b	sum	avg
5	5	34	6.8

Output



```
Enter the value of i(1 to 11)6
Enter the values67
Enter the values85
Enter the values4
Enter the values3
Enter the values62
Enter the values12
The sum of the values is233.000000
The average of the values is 38.8333
32
[Process completed (code 10) - press
 Enter]
```

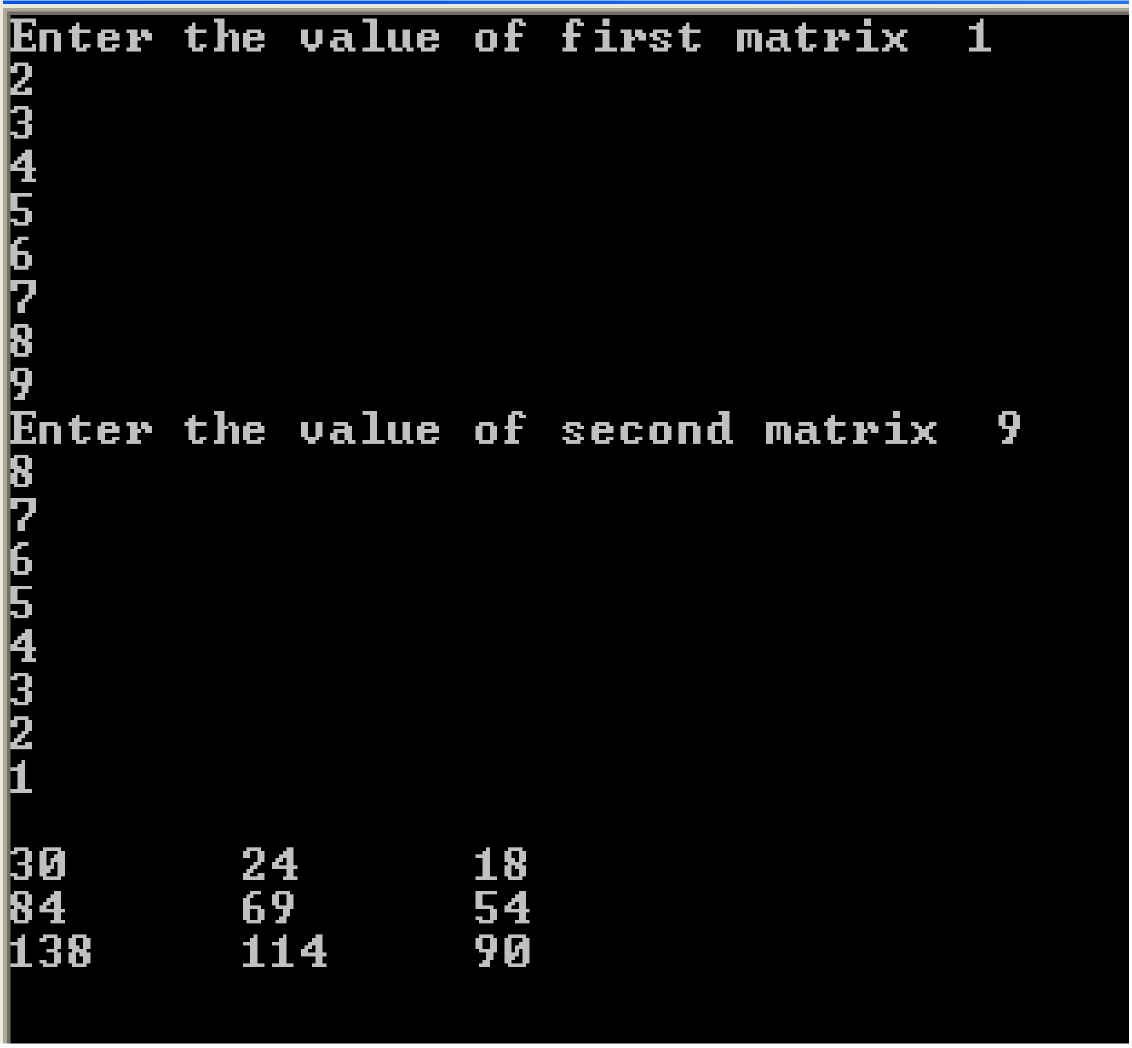
Result: The C program to perform arithmetic operation on 1-D Array was executed and output was verified

Practical 13: Write a C program to perform arithmetic operation on 2-D Array.
Aim: To write a C program to perform arithmetic operation on 2-D Array
Program:

```
// Matrix Multiplication
#include<stdio.h>
#include<conio.h>
```

```
void main()
{
 int a[3][3],b[3][3],c[3][3]={0,0,0,0,0,0,0,0,0},i,j,k;
clrscr();
printf("Enter the value of first matrix  ");
for(i=0;i<3;i++)
 {
for(j=0;j<3;j++)
  {
scanf("%d",&a[i][j]);
  }
 }
printf("Enter the value of second matrix  ");
for(i=0;i<3;i++)
 {
for(j=0;j<3;j++)
  {
scanf("%d",&b[i][j]);
  }
 }
for(i=0;i<3;i++)
 {
for(j=0;j<3;j++)
  {
for(k=0;k<3;k++)
   {
c[i][j]=a[i][k]*b[k][j]+c[i][j];
   }
  }
 }
for(i=0;i<3;i++)
 {
printf("\n");
for(j=0;j<3;j++)
  {
printf("%d",c[i][j]);
printf("\t");
  }
 }
getch();
}
```

Output



```
Enter the value of first matrix  1
2
3
4
5
6
7
8
9
Enter the value of second matrix  9
8
7
6
5
4
3
2
1

30       24       18
84       69       54
138      114      90
```

Result: The C program to perform arithmetic operation on 2-D Array was executed and output was verified

}

Practical 14: Write a C program to perform string manipulation function.

Aim: To write a C program to perform string manipulation function.

Few String Manipulation Functions
strlen(s) - calculates length of a string s
strcpy(s1,s) - copies a string s1 to another string s2
strcmp(s1,s2) - compares two strings s1 and s2
strcat(s1,s2) - concatenates two strings s1 and s2
strlwr(s)- converts s to lowercase
strupr(s)- converts s to uppercase
strncat(s1,s2,n)- first n characters of s2 is concatenated at the end of s1
strcmpi(s1,s2)- Compares s1 and s2 ignoring case
strncmpi(s1,s2)- Compares first n characters of s1 and s2 ignoring case
strncmp(s1,s2)- Compares first n characters of s1 and s2
strdup(s)- duplicates string s
strchr(s,c)-finds first occurrence of character c in string s
strrchr(s,c)-finds last occurrence of character c in string s
strstr(s1,s2)-finds first occurrence of string s2 in string s1
strset(s,c)-sets all the characters in the string s to c
strrev(s)-reverses the string

Program:

```
// C program for password checking using strcmp() function.
#include <stdio.h>
#include <string.h>
int main()
{
char password[]="PPS";
char input[15];
int match;
printf("Password: ");
scanf("%s",input);
match=strcmp(input,password);
if(match==0)
puts("Password accepted");
else
puts("Invalid password. Alert the authorities.");
return(0);
}
```

Output:

```
Password:PPS
Password accepted
```

Result: The C program to perform string manipulation function was executed and output was verified

Practical 15: Write a C program to search an element using linear or binary search.

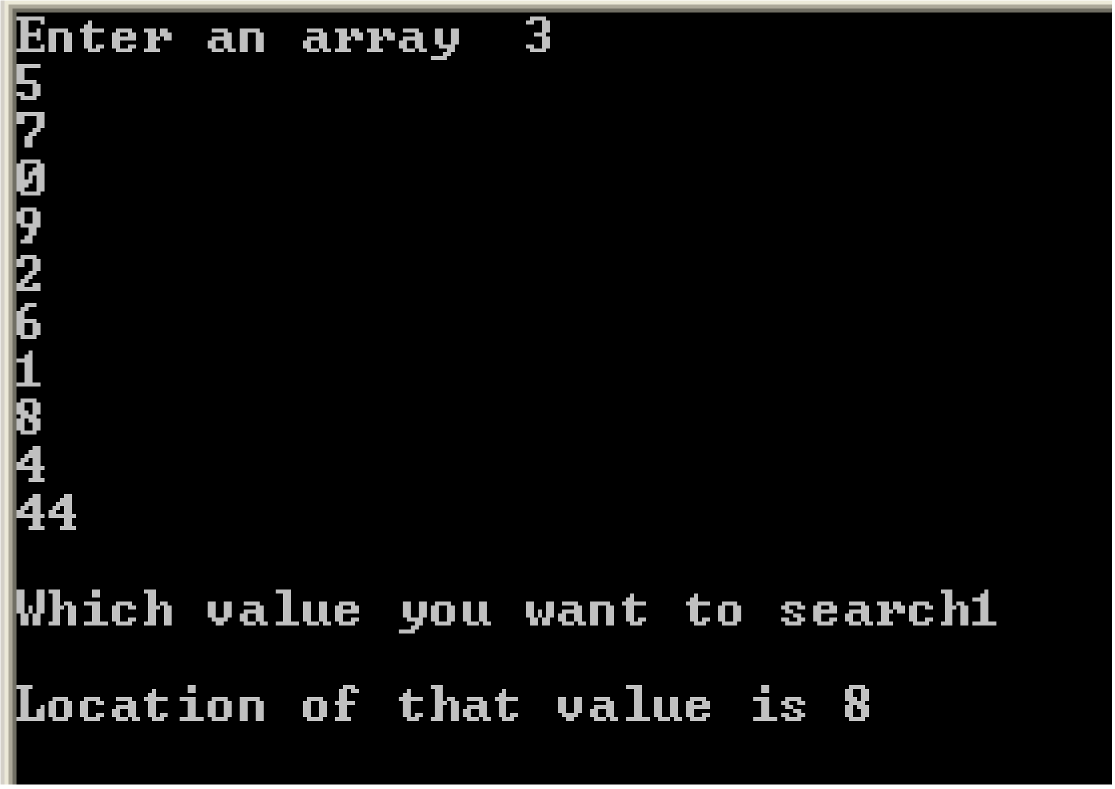
Aim: To write a C program to search an element using linear or binary search.

Program:

// Linear Search

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[11],b,c;
clrscr();
printf("Enter an array  ");
for(b=0;b<11;b++)
 {
scanf("%d",&a[b]);
 }
printf("\nWhich value you want to search");
scanf("%d",&c);
for(b=0;b<11;b++)
 {
if(a[b]==c)
  {
printf("\nLocation of that value is %d",b+1);
break;
  }
 }
if(b==11)
printf("\nValue not found in array");
getch();
}
```

Output



```
Enter an array  3
5
7
0
9
2
6
1
8
4
44

Which value you want to search1

Location of that value is 8
```

Result: The C program to search an element using linear search was executed and output was verified

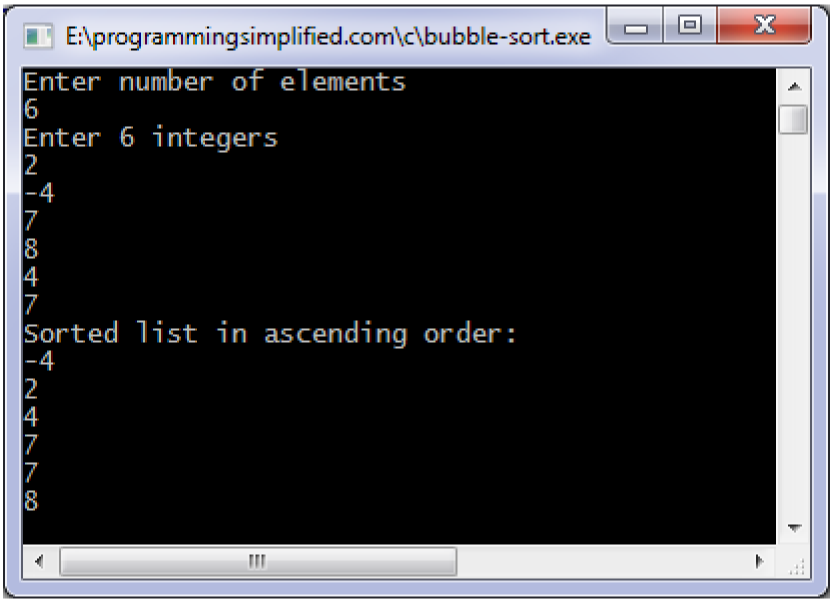
Practical 16: Write a C program to sort list of elements using bubble sort algorithm

Aim: To write a C program to sort list of elements using bubble sort algorithm**.**

<u>Program</u>

```
/* Bubble sort code */
 #include <stdio.h>
int main()
{
int array[100], n, c, d, swap;
printf("Enter number of elements\n");
scanf("%d", &n);
printf("Enter %d integers\n", n);
for (c = 0; c < n; c++)
scanf("%d", &array[c]);
for (c = 0 ; c < n - 1; c++)
  {
for (d = 0;d<n - c - 1; d++)
    {
if (array[d] > array[d+1]) /* For decreasing order use < */
      {
swap=array[d];
array[d]=array[d+1];
array[d+1]=swap;
      }
    }
  }
printf("Sorted list in ascending order:\n");
for (c = 0; c < n; c++)
printf("%d\n", array[c]);
return 0;
}
```

<u>Output of program:</u>



```
Enter number of elements
6
Enter 6 integers
2
-4
7
8
4
7
Sorted list in ascending order:
-4
2
4
7
7
8
```

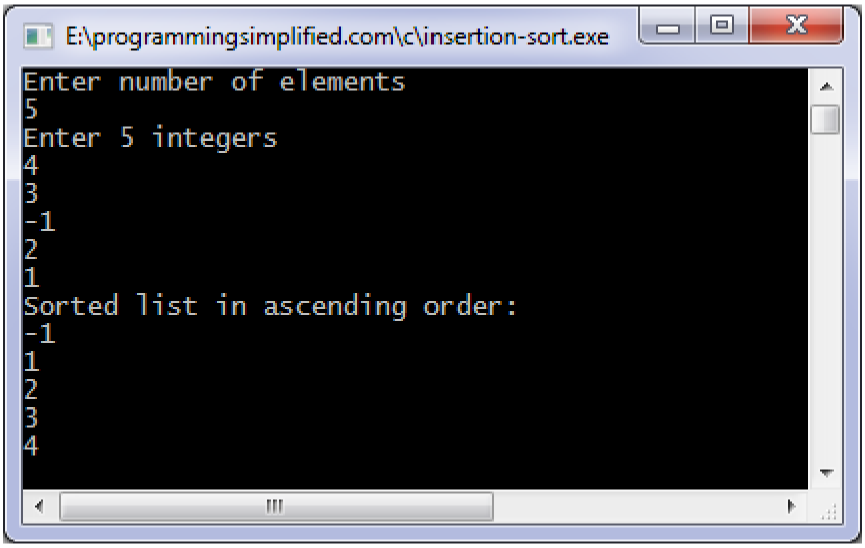
Result: The C program to sort list of elements using bubble sort algorithmwas executed and output was verified

Practical 17: Write a C program to sort list of elements using Insertion sort algorithm

Aim: To write a C program to sort list of elements using insertion sort algorithm**.**

Program

```
/* Insertion sort ascending order */
#include <stdio.h>
int main()
{
int n,A[1000],i,j,key;
printf("Enter number of elements\n");
scanf("%d", &n);
printf("Enter %d integers\n", n);
for (i=0;i<n;i++) {
scanf("%d", &A[i]);
  }
for (i=1;i< n ; i++)
{
key=A[i ];
 j=i-1;
while(j > =0 && A [ j ] > key)
{
A[j+1]=A[j];
j--;
 }
A [ j+1 ] = key;
}
printf("Sorted list in ascending order:\n");

for (i= 0;i<n;i++) {
printf("%d\n",A[i]);
  }
return 0;
}
```



```
Enter number of elements
5
Enter 5 integers
4
3
-1
2
1
Sorted list in ascending order:
-1
1
2
3
4
```

Result: The C program to sort list of elements using insertion sort algorithmwas executed and output was verified.

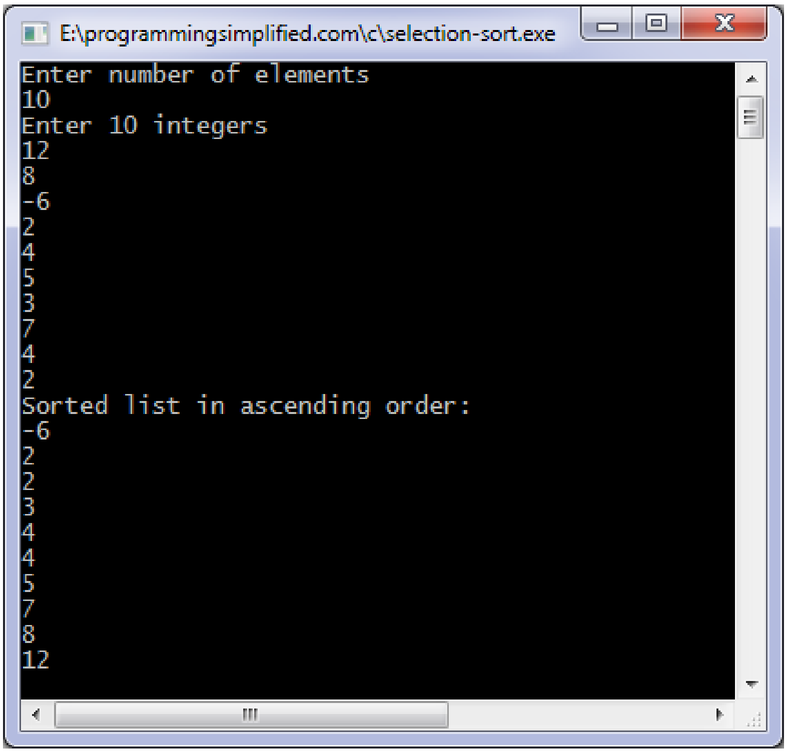
Practical 18: Write a C program to sort list of elements using Selection sort algorithm

Aim: To write a C program to sort list of elements using Selection sort algorithm**.**

Program

```
/* Selection sort ascending order */
#include <stdio.h>
int main()
{
int array[100], n, c, d, position, swap;
printf("Enter number of elements\n");
scanf("%d", &n);
printf("Enter %d integers\n", n);

for ( c = 0 ; c < n ; c++ )
scanf("%d", &array[c]);
for ( c = 0 ; c < ( n - 1 ) ; c++ )
   {
position = c;
for ( d = c + 1 ; d < n ; d++ )
      {
if ( array[position] > array[d] )
position = d;
      }
if ( position != c )
      {
swap = array[c];
array[c] = array[position];
array[position] = swap;
      }
   }
printf("Sorted list in ascending order:\n");
for ( c = 0 ; c < n ; c++ )
printf("%d\n", array[c]);
return 0;
}
```

Output:



```
Enter number of elements
10
Enter 10 integers
12
8
-6
2
4
5
3
7
4
2
Sorted list in ascending order:
-6
2
2
3
4
4
5
7
8
12
```

Result: The C program to sort list of elements using selection sort algorithmwas executed and output was verified.

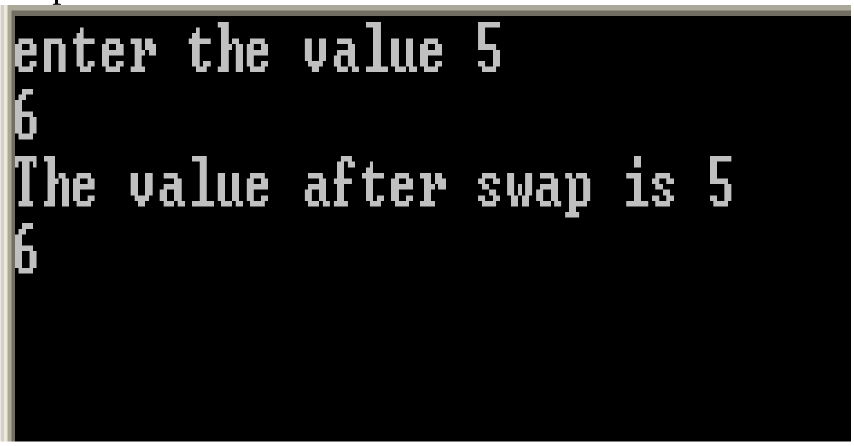
Practical 19: Write a C program to swap two numbers using call by value and call by reference.

Aim: To write a C program to swap two numbers using call by value and call by reference.

<u>Program</u>

```
//Call by Value
#include<stdio.h>
#include<conio.h>
void swap(int i,int j)
{
int temp=i;
i=j;
j=temp;
}
void main()
{
int a,b;
clrscr();
printf("enter the value ");
scanf("%d%d",&a,&b);
swap(a,b);
printf("The value after swap is %d\n%d",a,b);
getch();
}
```

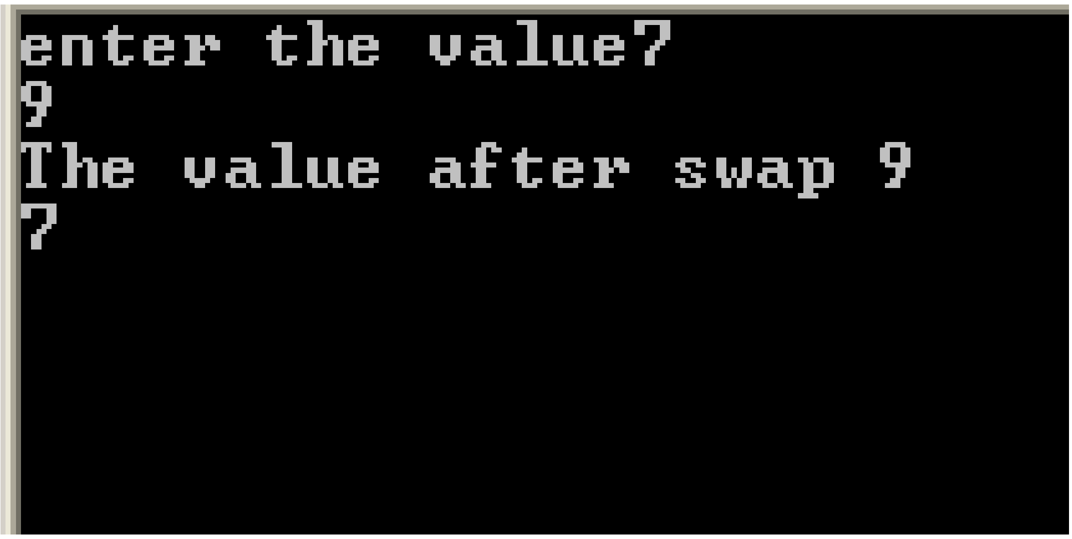
Output



```
enter the value 5
6
The value after swap is 5
6
```

```
//Call by Reference
#include<stdio.h>
#include<conio.h>
void swap(int *i,int *j)
{
int temp=(*i);
(*i)=(*j);
(*j)=temp;
}
void main()
{
int a,b;
clrscr();
printf("enter the value ");
scanf("%d%d",&a,&b);
swap(&a,&b);
printf("The value after swap %d\n%d",a,b);
getch();
}
```

Output



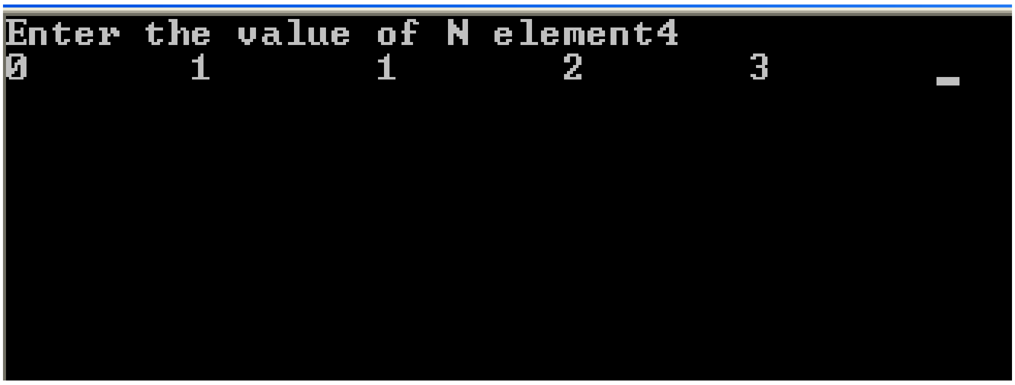
```
enter the value7
9
The value after swap 9
7
```

Result: The C program to swap two numbers using call by value and call by reference was executed and output was verified.

Practical 20: Write a C program to find Fibonacci series using recursion.
Aim: To write a C program to find Fibonacci series using recursion.
<u>Program:</u>

```
//Fibonacci Series using Recursion

#include<stdio.h>
void Fib(int n){
static int n1=0,n2=1,n3;
if(n>0){
        n3 = n1 + n2;
        n1 = n2;
        n2 = n3;
printf("%d ",n3);
Fib(n-1);
    }
}
int main(){
int n;
printf("Enter the number of elements: ");
scanf("%d",&n);
printf("Fibonacci Series: ");
printf("%d %d ",0,1);
Fib(n-2);//n-2 because 2 numbers are already printed
return 0;
 }
```

<u>Output</u>



```
Enter the value of N element4
0       1       1       2       3       _
```

Result: The C program to find Fibonacci series using recursion was executed and output was verified.

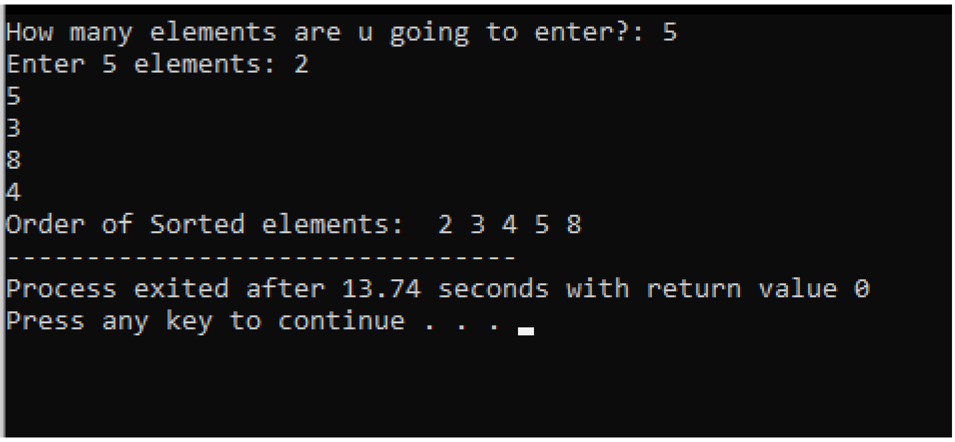
Practical 21: Write a C program to sort list of elements using Quick sort.

Aim: To write a C program to sort list of elements using Quick sort.

Program:

```
#include<stdio.h>
void quicksort(int number[25],int first,int last)
{
int i, j, pivot, temp;
if(first<last){
pivot=first;
i=first;
    j=last;
while(i<j)
{
while(number[i]<=number[pivot]&&i<last)
i++;
while(number[j]>number[pivot])
j--;
if(i<j){
temp=number[i];
number[i]=number[j];
number[j]=temp;
      }
    }
temp=number[pivot];
number[pivot]=number[j];
number[j]=temp;
quicksort(number,first,j-1);
quicksort(number,j+1,last);
  }
}
int main(){
int i, count, number[25];
printf("How many elements are u going to enter?: ");
scanf("%d",&count);
printf("Enter %d elements: ", count);
for(i=0;i<count;i++)
scanf("%d",&number[i]);
quicksort(number,0,count-1);
printf("Order of Sorted elements: ");
for(i=0;i<count;i++)
printf(" %d",number[i]);
return 0;
}
```

Output:



```
How many elements are u going to enter?: 5
Enter 5 elements: 2
5
3
8
4
Order of Sorted elements:  2 3 4 5 8
--------------------------------
Process exited after 13.74 seconds with return value 0
Press any key to continue . . .
```

Result: The C program to sort list of elements using Quick sort was executed and output was verified.

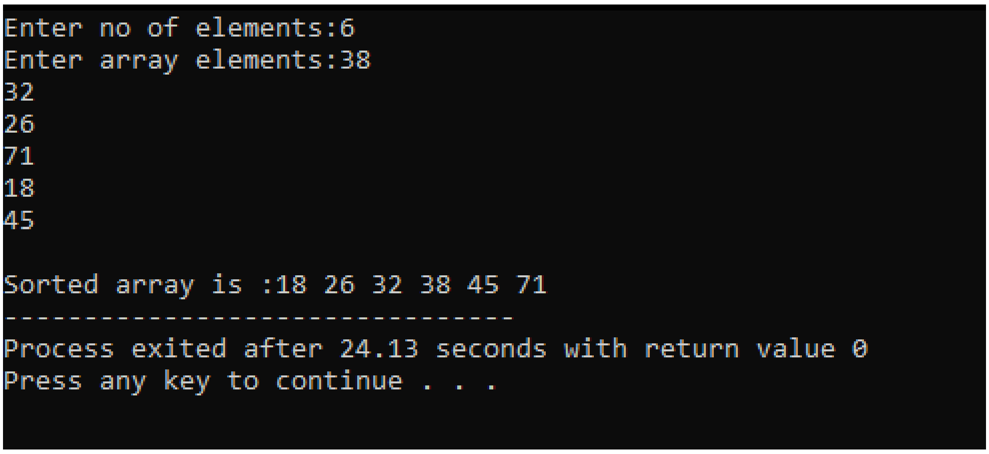
Practical 22: Write a C program to sort list of elements using Merge sort..

Aim: To write a C program to sort list of elements using Merge sort..

Program:

```
#include<stdio.h>
void mergesort(int a[],int i,int j);
void merge(int a[],int i1,int j1,int i2,int j2);
int main()
{
int a[30],n,i;
printf("Enter no of elements:");
scanf("%d",&n);
printf("Enter array elements:");

for(i=0;i<n;i++)
scanf("%d",&a[i]);

mergesort(a,0,n-1);

printf("\nSorted array is :");
for(i=0;i<n;i++)
printf("%d ",a[i]);

return 0;
}

void mergesort(int a[],int i,int j)
{
int mid;

if(i<j)
   {
mid=(i+j)/2;
mergesort(a,i,mid);        //left recursion
mergesort(a,mid+1,j);    //right recursion
merge(a,i,mid,mid+1,j);    //merging of two sorted sub-arrays
   }
}

void merge(int a[],int i1,int j1,int i2,int j2)
{
int temp[50];    //array used for merging
int i,j,k;
i=i1;    //beginning of the first list
   j=i2;    //beginning of the second list
   k=0;
while(i<=j1 && j<=j2)    //while elements in both lists
    {
if(a[i]<a[j])
temp[k++]=a[i++];
else
temp[k++]=a[j++];
    }

while(i<=j1)    //copy remaining elements of the first list
temp[k++]=a[i++];

while(j<=j2)    //copy remaining elements of the second list
temp[k++]=a[j++];

    //Transfer elements from temp[] back to a[]
for(i=i1,j=0;i<=j2;i++,j++)
a[i]=temp[j];
}
```

Output:



```
Enter no of elements:6
Enter array elements:38
32
26
71
18
45

Sorted array is :18 26 32 38 45 71
--------------------------------
Process exited after 24.13 seconds with return value 0
Press any key to continue . . .
```

Result: The C program to sort list of elements using merge sort was executed and output was verified.

Practical 23: Write a C program to perform Array of Structure.

Aim: To write a C program to perform Array of Structure.

<u>Program:</u>

```
#include<stdio.h>
#include<string.h>
#define MAX 2
struct student
{
char name[20];
int roll_no;
float marks;
};
int main()
{
struct student arr_student[MAX];
int i;
for(i = 0; i< MAX; i++ )
   {
printf("\nEnter details of student %d\n\n", i+1);
printf("Enter name: ");
scanf("%s", arr_student[i].name);
printf("Enter roll no: ");
scanf("%d", &arr_student[i].roll_no);
printf("Enter marks: ");
scanf("%f", &arr_student[i].marks);
   }
printf("\n");
printf("Name\tRoll no\tMarks\n");
for(i = 0; i< MAX; i++ )
   {
printf("%s\t%d\t%.2f\n",
arr_student[i].name, arr_student[i].roll_no, arr_student[i].marks);
   }
return 0;
}
```

Output:

```
Enter details of student 1
Enter name: Jim
Enter roll no: 1
Enter marks: 44
Enter details of student 2
Enter name: Sim
Enter roll no: 2
Enter marks: 76
Name Roll no Marks
Jim  1      44.00
```

```
    Sim  2       76.00
```

Result: The C program to perform Array of Structure was executed and output was verified.

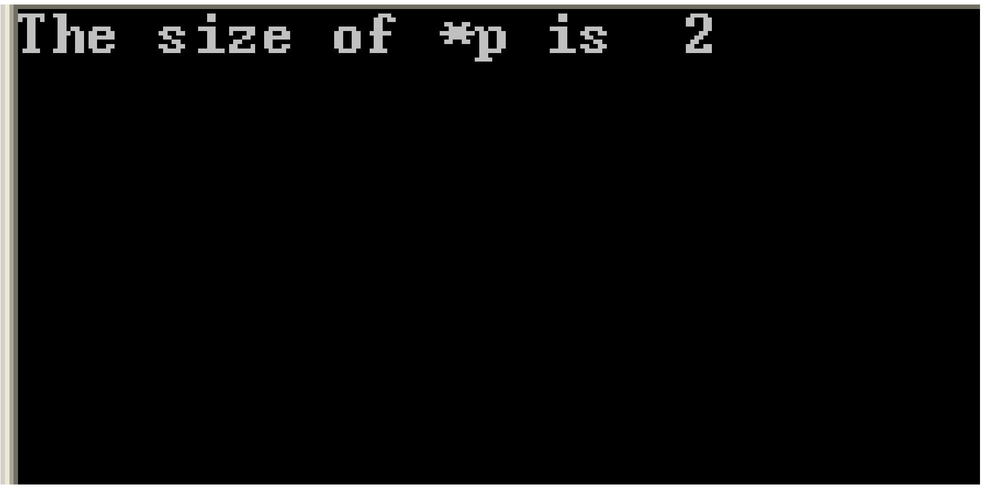
Practical 24: Write a C program to implement pointers.

Aim: To write a C program to implement pointers.

Program:

```
#include<stdio.h>
int main()
{
int *p;
printf("The size of *p is  %d",sizeof(p));
return 0;
}
```

Output:



```
The size of *p is  2
```

Result: The C program to implement pointers was executed and output was verified.

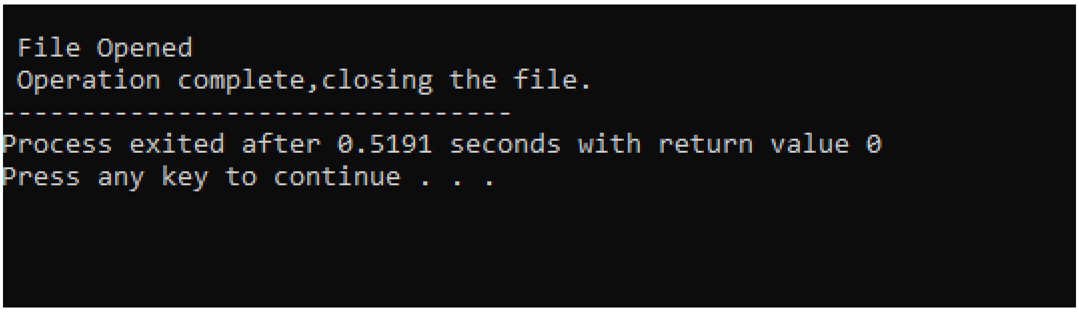
Practical 25: Write a C program to perform file handling.

Aim: To write a C program to perform file handling..

Program:

```
#include<stdio.h>
int main()
{
FILE *f;
f=fopen("Test.txt","w");
if(f==NULL)
{
printf("\n Sorry,File cannot be opened");
printf("\n Program Terminating...");
}
else
{
printf("\n File Opened");
fprintf(f,"%s","Test Successful !");
printf("\n Operation complete,closing the file.");
fclose(f);
}
    return 0;
}
```

Output:



```
 File Opened
 Operation complete,closing the file.
--------------------------------
Process exited after 0.5191 seconds with return value 0
Press any key to continue . . .
```

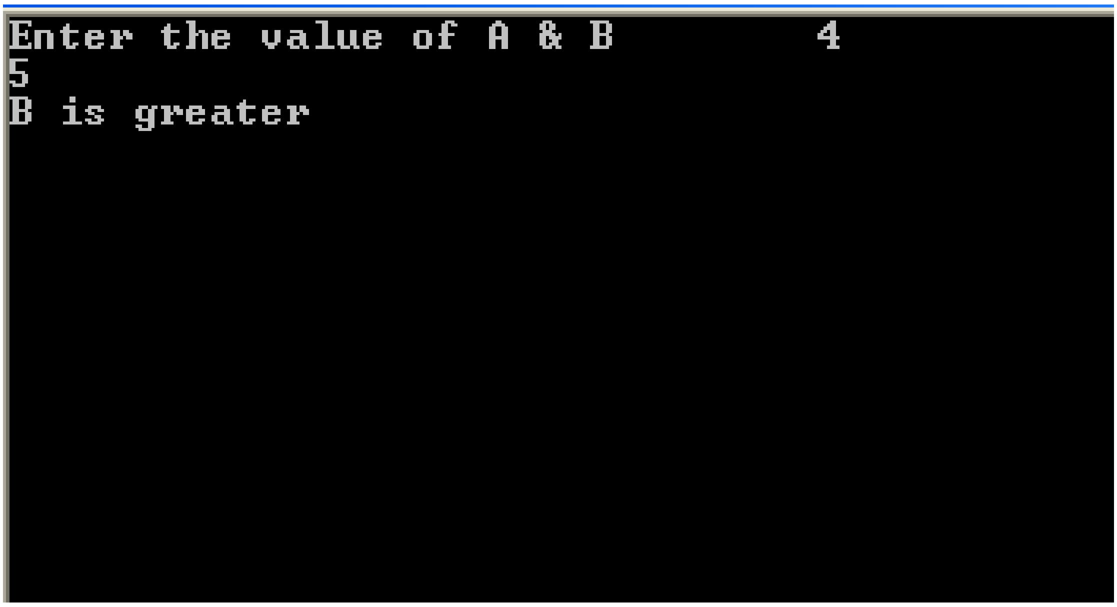
Result: The C program to perform file handling was executed and output was verified.

Practical beyond Syllabus:

1. **Find the largest of 2 no. using condition operator**

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
clrscr();
printf("Enter the value of A & B\t");
scanf("%d%d",&a,&b);
 (a>b)?printf("A is greater"):printf("B is greater");
getch();
}
```

Output

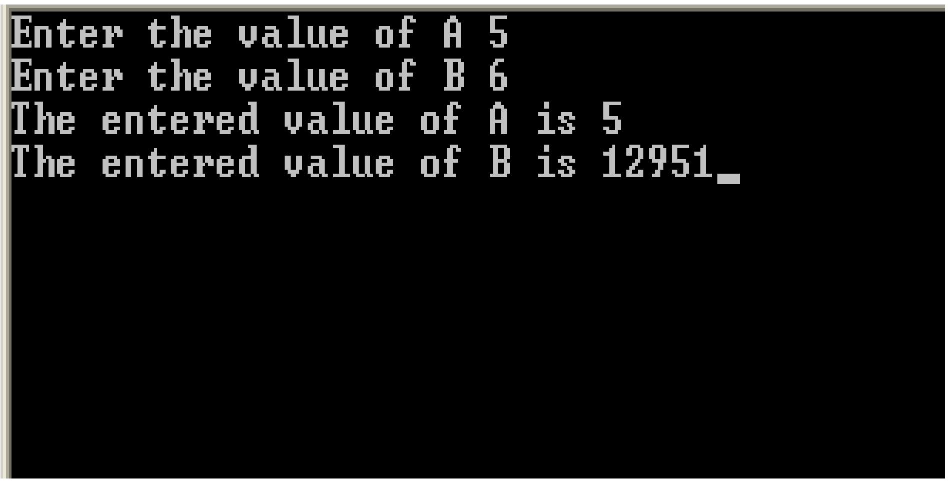


```
Enter the value of A & B        4
5
B is greater
```

2. To demonstrate the importance of ampersand(&)operator in scanf

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
clrscr();
printf("Enter the value of A ");
scanf("%d",&a);
printf("Enter the value of B ");
scanf("%d",b);
printf("The entered value of A is %d\n",a);
printf("The entered value of B is %d",b);
getch();
}
```

Output

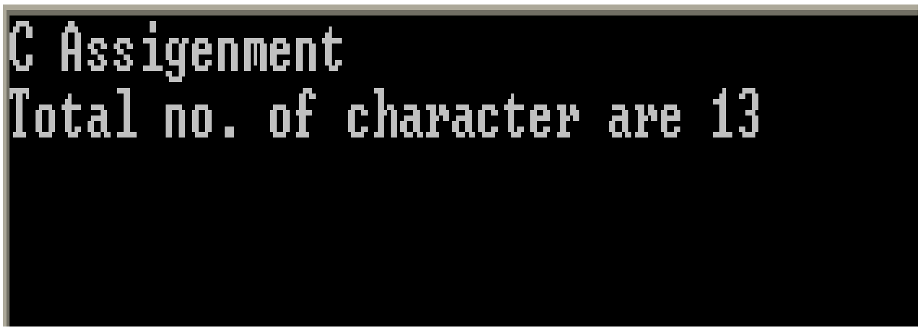


```
Enter the value of A 5
Enter the value of B 6
The entered value of A is 5
The entered value of B is 12951_
```

3. To demonstrate the work of %n

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
clrscr();
printf("C Assigenment%n\n",&a);
 /*%n is used to count the character*/
printf("Total no. of character are %d",a);
getch();
}
```

Output

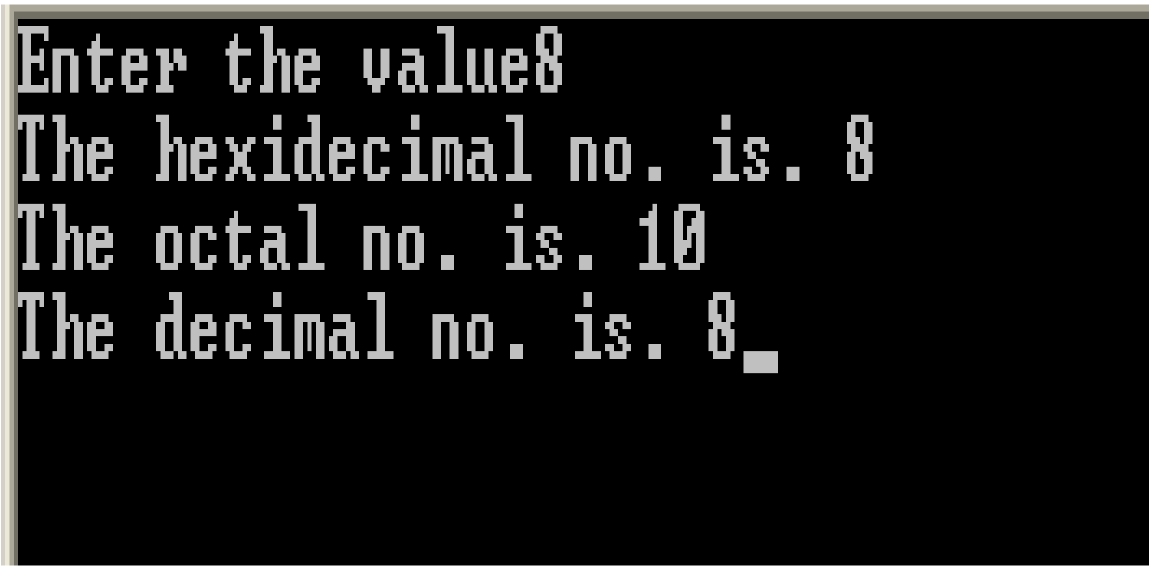


```
C Assigenment
Total no. of character are 13
```

4. Display decimal, octal & hexadecimal no.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
clrscr();
printf("Enter the value");
scanf("%d",&a);
printf("The hexidecimal no. is. %x",a);
printf("\nThe octal no. is. %o",a);
printf("\nThe decimal no. is. %d",a);
getch();
}
```

Output

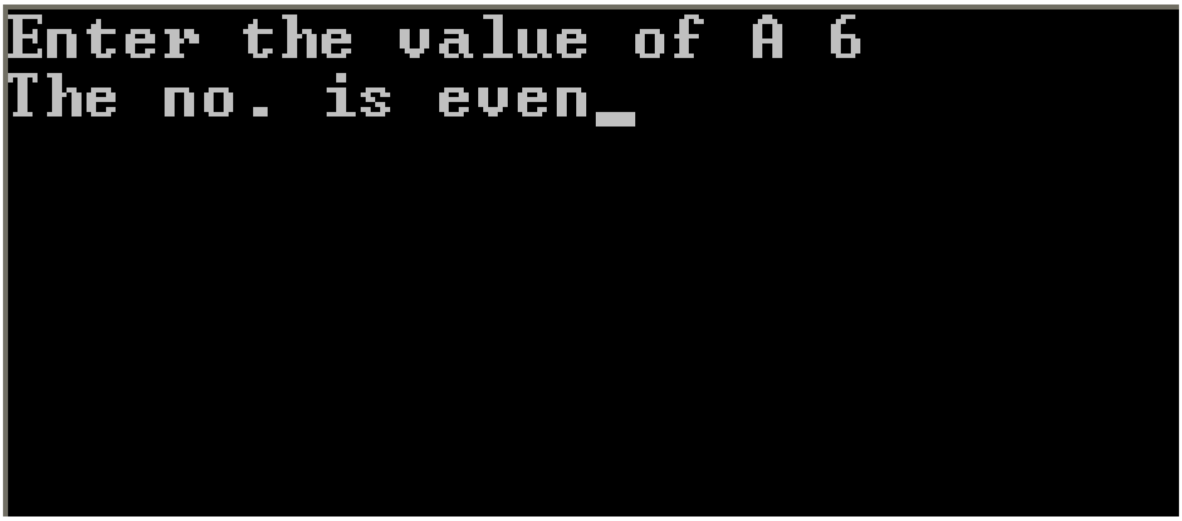


```
Enter the value8
The hexidecimal no. is. 8
The octal no. is. 10
The decimal no. is. 8_
```

5. Check the no. is positive or negative

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
clrscr();
printf("Enter the value");
scanf("%d",&a);
if(a>0)
printf("Entered no. is positive");
else if(a<0)
printf("Entered no. is negative");
else
printf("Entered no. is zero");
getch();
}
```

Output

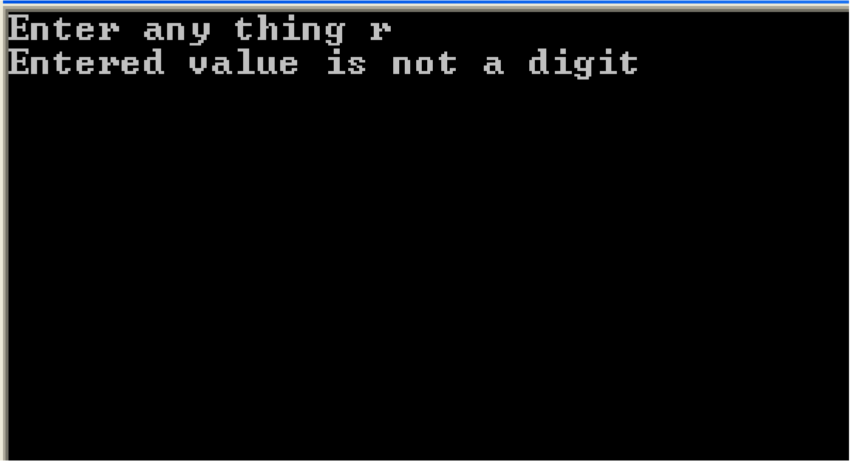


```
Enter the value of A 6
The no. is even_
```

6. Given character is digit or not

```
#include<stdio.h>
#include<conio.h>
void main()
{
char ch;
clrscr();
printf("Enter any thing ");
scanf("%c",&ch);
if(isdigit(ch))
printf("Entered value is digit");
else
printf("Entered value is not a digit");
getch();
 }
```

Output

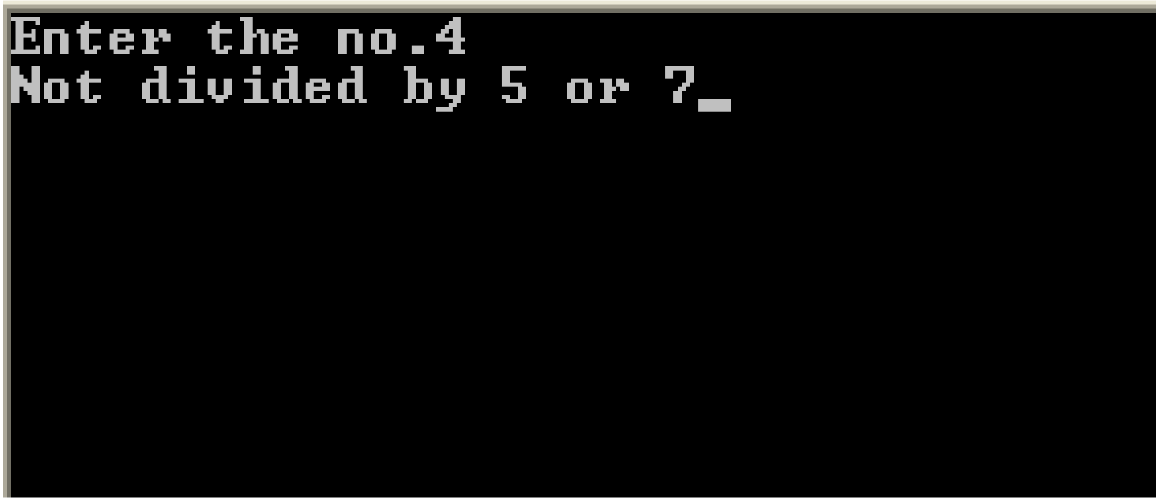


```
Enter any thing r
Entered value is not a digit
```

7. Check the no.is divided by 5 or 7 or by both or by no one

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
clrscr();
printf("Enter the no.");
scanf("%d",&a);
if(a%5==0&&a%7==0)
printf("Entered no. is divided by 5 & 7");
else if(a%5==0)
printf("Entered no. is divided by 5");
else if(a%7==0)
printf("Entered no. is divided by 7");
else
printf("Not divided by 5 or 7");
getch();
}
```

Output

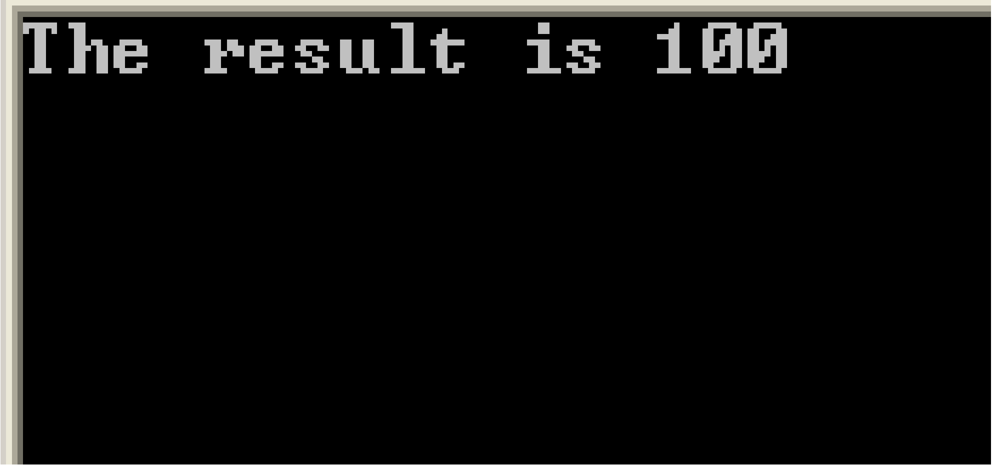


```
Enter the no.4
Not divided by 5 or 7_
```

8. To demonstrate the working of preprocessor

```
#include<stdio.h>
#include<conio.h>
#define N 10
void main()
{
clrscr();
printf( "The result is %d\n",N*N);
getch();
}
```

Output

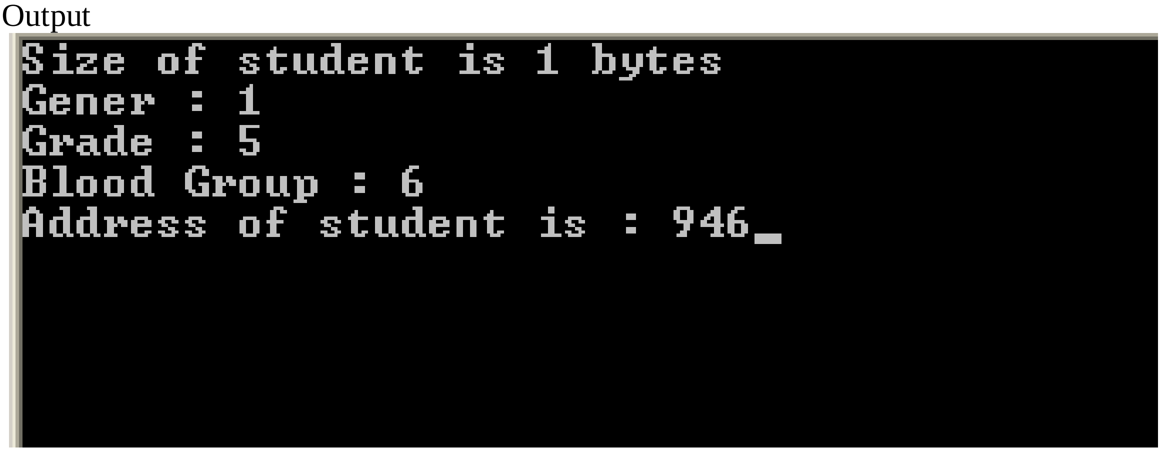


```
The result is 100
```

9. To demonstrate the use of bitfields

```
#include<stdio.h>
#include<conio.h>
struct
{
unsigned gender: 1;
unsigned grade: 4;
unsigned bloodgroup:3;
}s1;
void main()
{
clrscr();
printf("Size of student is %d bytes",sizeof(s1));
s1.gender=1;
s1.grade=5;
s1.bloodgroup=6;
printf("\nGener : %d\nGrade : %d\nBlood Group : %d"
,s1.gender,s1.grade,s1.bloodgroup);
printf("\nAddress of student is : %u",&s1);
getch();
}
```

Output

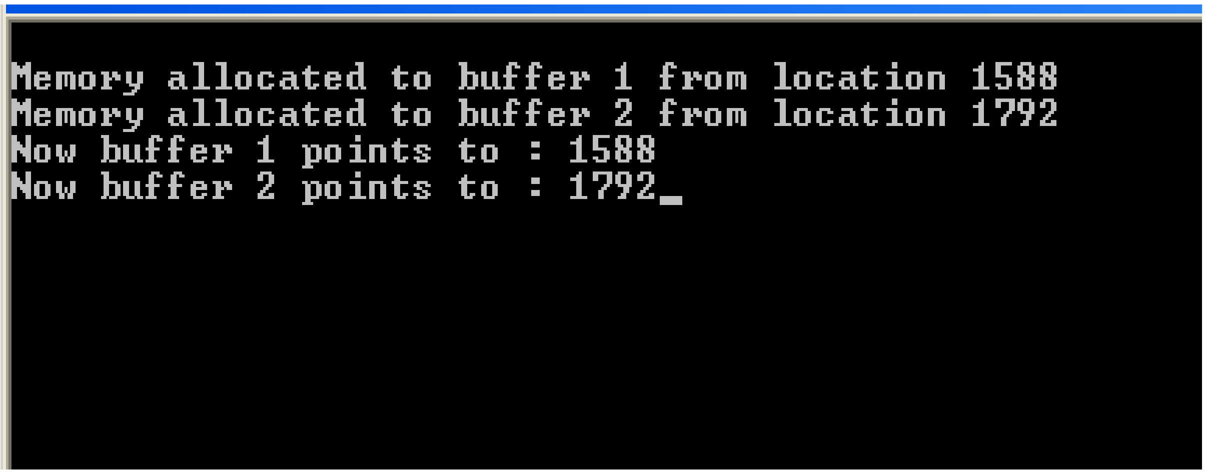


```
Size of student is 1 bytes
Gener : 1
Grade : 5
Blood Group : 6
Address of student is : 946_
```

10.Demonstrate the use of mallock, callockand free function

```
#include<stdio.h>
#include<conio.h>
void main()
{
int *b1,*b2;
clrscr();
b1=(int*)malloc(100*sizeof(int));
b2=(int*)calloc(100,sizeof(int));
printf("\nMemory allocated to buffer 1 from location %u",b1);
printf("\nMemory allocated to buffer 2 from location %u",b2);
free(b1);
free(b2);
printf("\nNow buffer 1 points to : %u",b1);
printf("\nNow buffer 2 points to : %u",b2);
getch();
}
```

Output

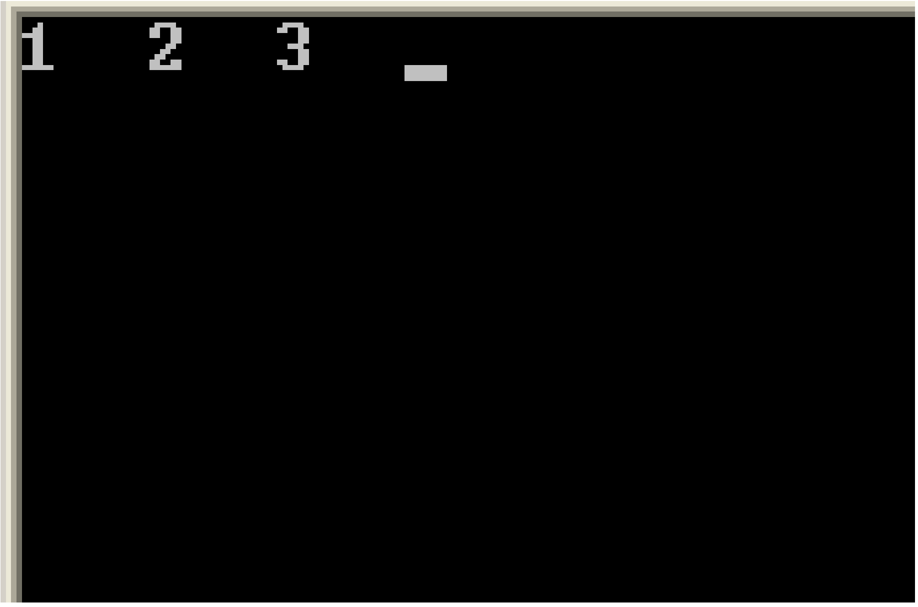


```
Memory allocated to buffer 1 from location 1588
Memory allocated to buffer 2 from location 1792
Now buffer 1 points to : 1588
Now buffer 2 points to : 1792_
```

11.To explain the use of static storage classes

```
#include<stdio.h>
#include<conio.h>
void change()
{
static int i;
i++;
printf("%d  ",i);
}
void main()
{
clrscr();
change();
change();
change();
getch();
}
```

Output

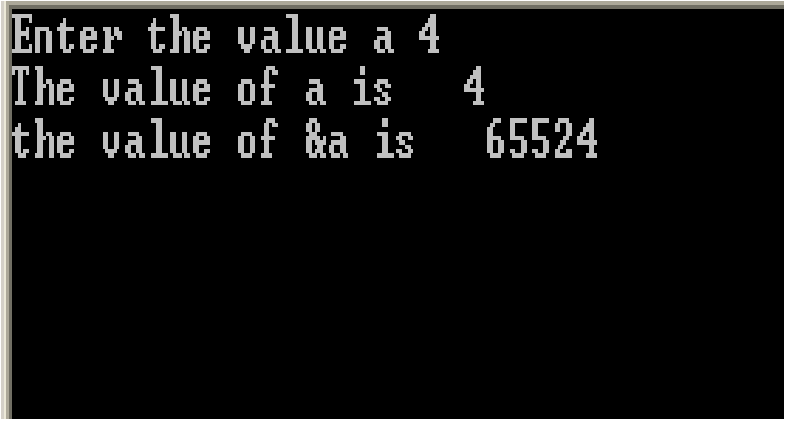


```
1    2    3    _
```

12.To show different between a and &a

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
clrscr();
printf("Enter the value a ");
scanf("%d",&a);
printf("The value of a is   %d",a);
printf("\nthe value of &a is   %u",&a);
getch();
}
```

Output

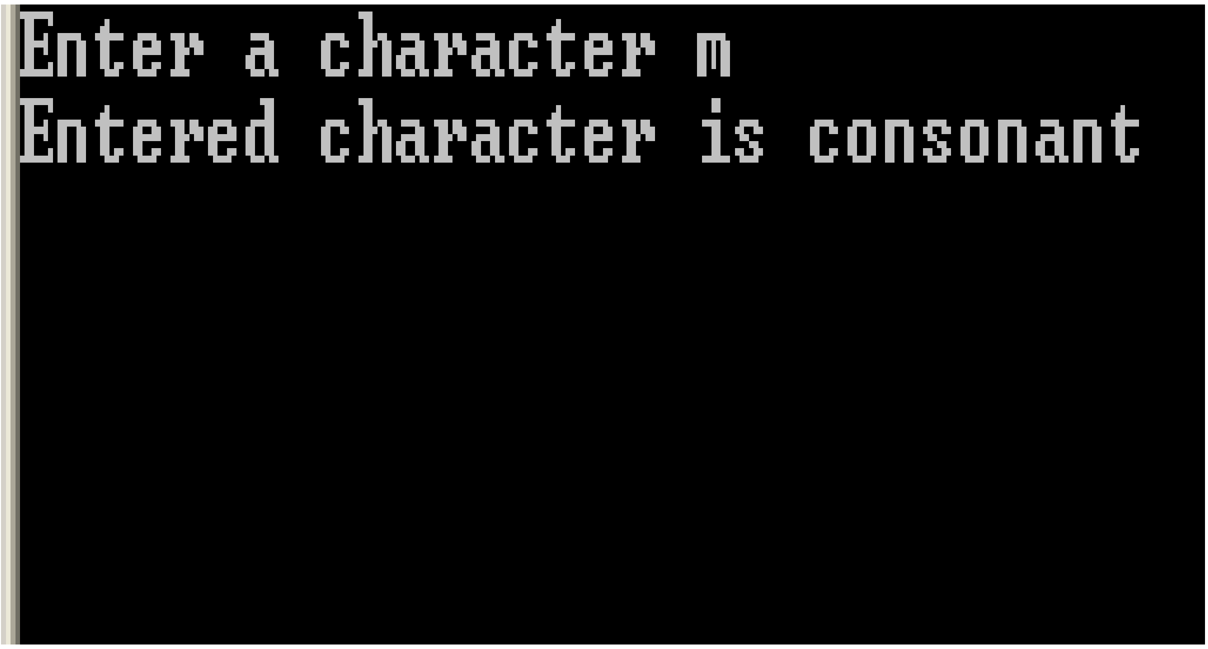


```
Enter the value a 4
The value of a is   4
the value of &a is   65524
```

13. Check the value is vowal or consonant

```
#include<stdio.h>
#include<conio.h>
void main()
{
char ch;
clrscr();
printf("Enter a character ");
scanf("%c",&ch);
switch(ch)
 {
case 'A':
case 'a':
case 'E':
case 'e':
case 'I':
case 'I':
case 'O':
case 'o':
case 'U':
case 'u':
printf("Entered character is vowal");
break;
default:
printf("Entered character is consonant");
break;
 }
getch();
}
```

Output

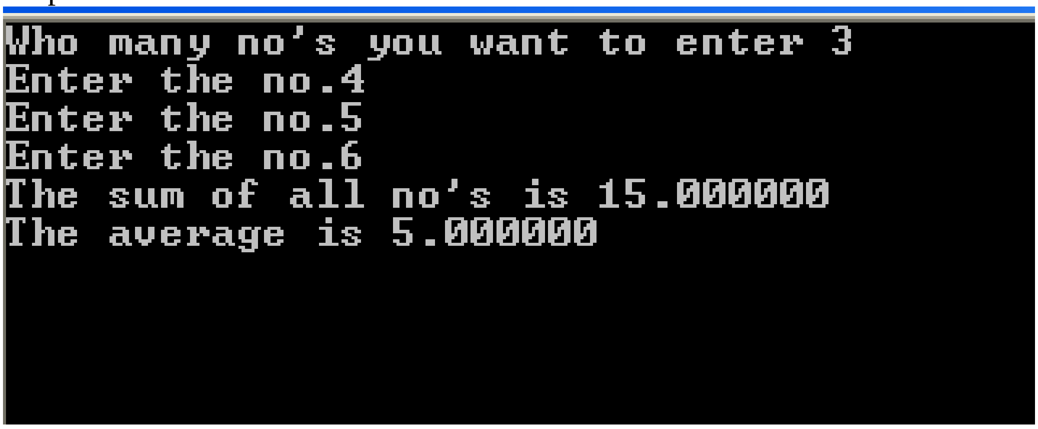


```
Enter a character m
Entered character is consonant
```

14. Read N no. from user and find average

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,i,a;
float avg,sum=0;
clrscr();
printf("Who many no's you want to enter ");
scanf("%d",&n);
for(i=1;i<=n;i++)
 {
printf("Enter the no.");
scanf("%d",&a);
sum=sum+a;
 }
printf("The sum of all no's is %f\n",sum);
avg=sum/n;
printf("The average is %f",avg);
getch();
}
```

Output

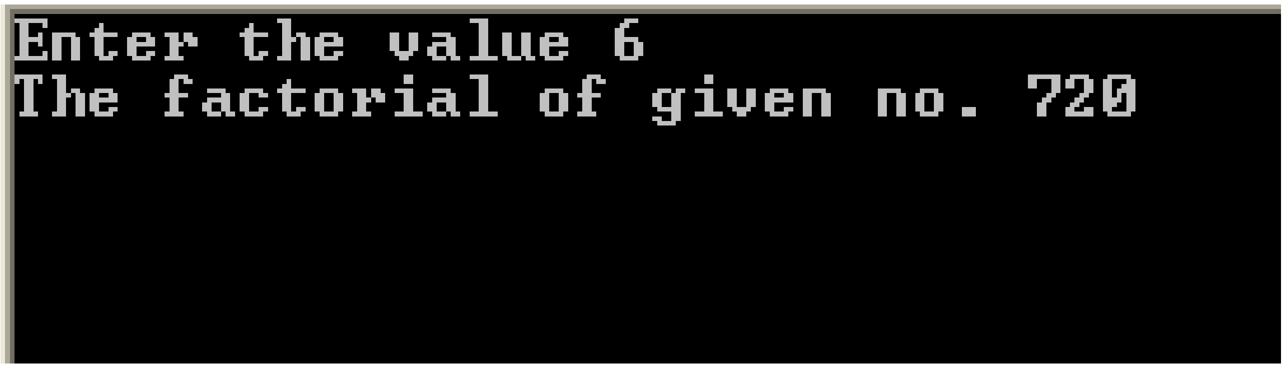


```
Who many no's you want to enter 3
Enter the no.4
Enter the no.5
Enter the no.6
The sum of all no's is 15.000000
The average is 5.000000
```

15. Factorial of given no.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,f;
clrscr();
printf("Enter the value ");
scanf("%d",&a);
 f=a-1;
while(f>1)
  {
   a=a*f--;
  }
printf("The factorial of given no. %d",a);
getch();
}
```

Output

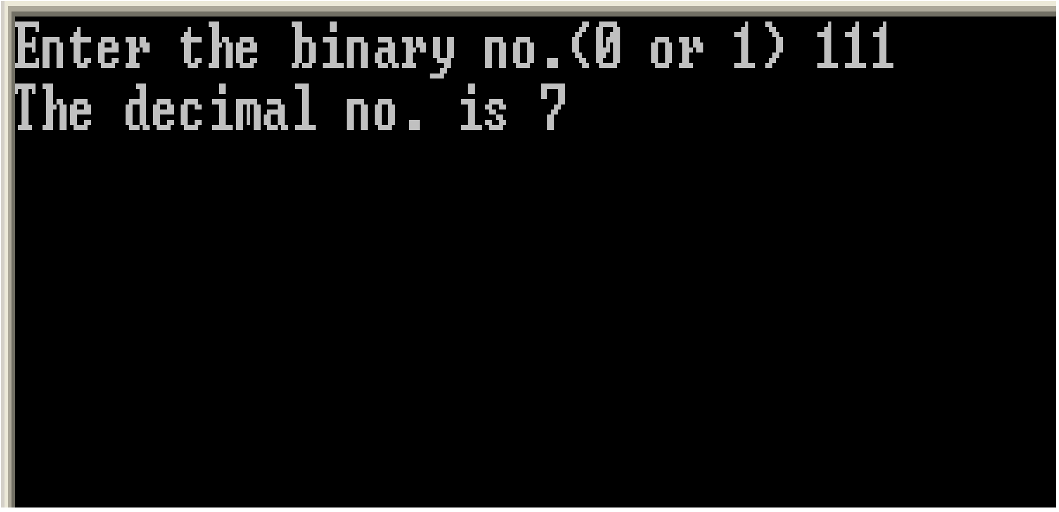


```
Enter the value 6
The factorial of given no. 720
```

16. Convert binary no. to decimal no.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
long int num,d,p=0,r=0;
clrscr();
printf("Enter the binary no.(0 or 1) ");
scanf("%ld",&num);
while(num>0)
 {
  d=num%10;
num=num/10;
  r=r+d*pow(2,p);
p++;
 }
printf("The decimal no. is %ld",r);
getch();
}
```

Output

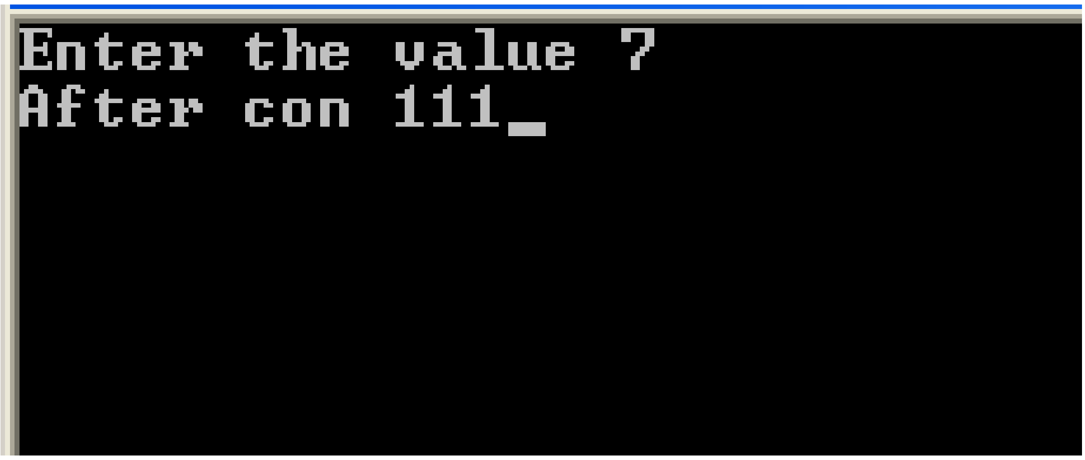


```
Enter the binary no.(0 or 1) 111
The decimal no. is 7
```

17. Convert decmal no. to binary no.

```
#include<stdio.h>
#include<conio.h>
void main()
{
long unsigned int num,sum=0,i=1,x;
clrscr();
printf("Enter the value ");
scanf("%ld",&num);
while(num>0)
 {
  x=num%2;
sum=x*i+sum;
i=i*10;
num=num/2;
 }
printf("After con %ld",sum);
getch();
}
```

Output

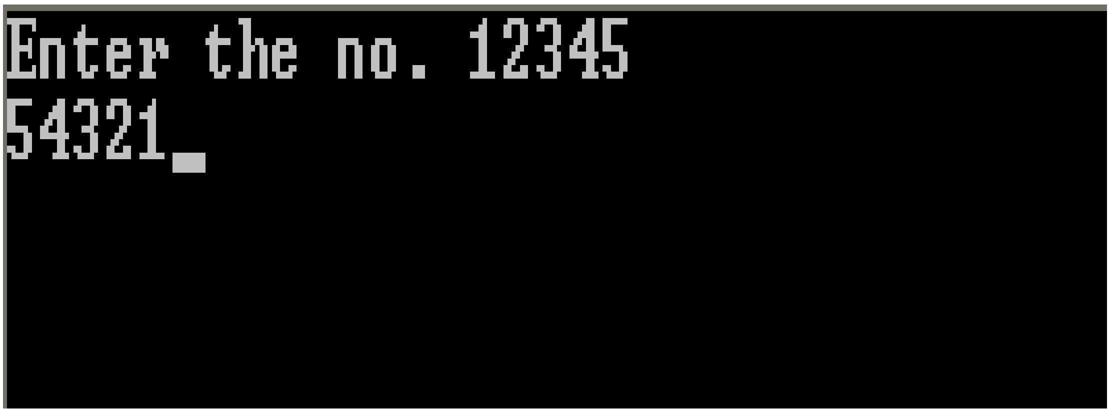


```
Enter the value 7
After con 111_
```

18. Reverse of no.

```
#include<stdio.h>
#include<conio.h>
void main()
{
long int a,b;
clrscr();
printf("Enter the no. ");
scanf("%ld",&a);
while(a>0)
 {
  b=a%10;
  a=a/10;
printf("%ld",b);
 }
getch();
}
```

Output

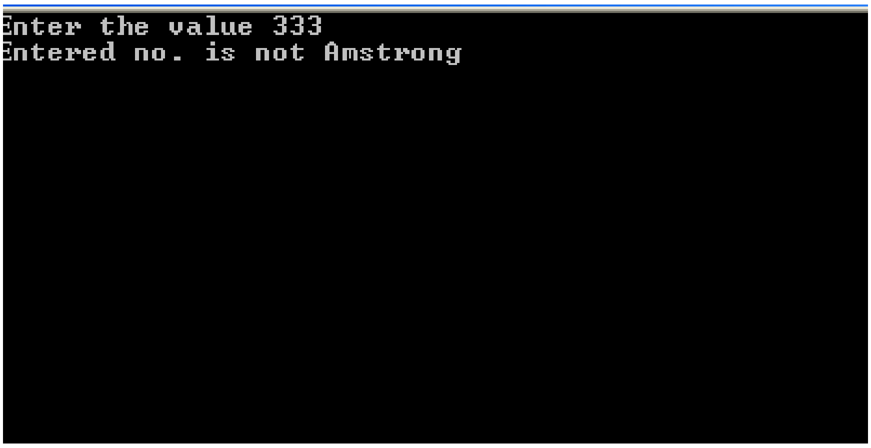


```
Enter the no. 12345
54321_
```

19. Check the no. is amstrong or not

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
int a,b,c,num,count=0,sum=0,x;
clrscr();
printf("Enter the value ");
scanf("%d",&num);
 a=b=num;
while(num>0)
 {
  x=num%10;
num=num/10;
count++;
 }
while(b>0)
 {
  c=b%10;
sum=sum+pow(c,count);
  b=b/10;
 }
if(a==sum)
printf("Entered no. is Amstrong");
else
printf("Entered no. is not Amstrong");
getch();
}
```

Output

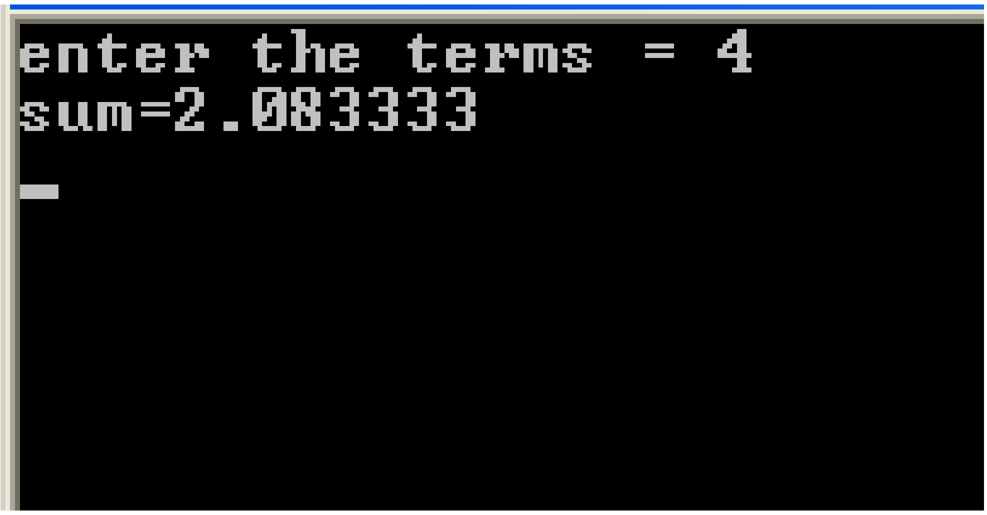


```
Enter the value 333
Entered no. is not Amstrong
```

20. To find the sum of series 1+ $^{1}/_{2}$ + $^{1}/_{3}$+ $^{1}/_{4}$+...... $^{1}/_{n}$

```
#include <stdio.h>
#include <conio.h>
void main ()
{
float i,n;
float sum =0;
clrscr();
printf ("enter the terms = ");
scanf ("%f",&n);
for (i=1;i<=n;i++)
   {
sum += 1/i;
   }
printf ("sum=%f\n",sum);
getch();
}
```

Output

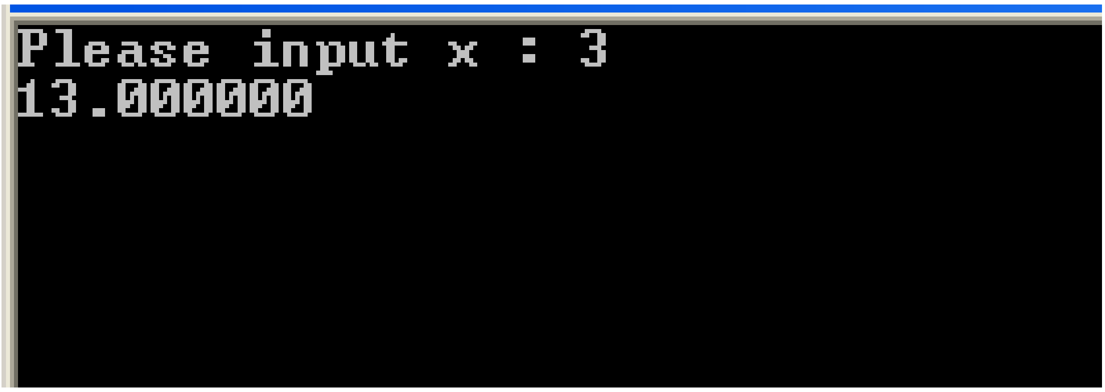


```
enter the terms = 4
sum=2.083333
```

21. To find the sum of series 1+x $+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\cdots+\frac{x^n}{n!}$

```
#include <stdio.h>
#include <math.h>
int factorial(int n)
{
if(n==1)
return 1;
else
return (n * factorial(n-1));
}
int main()
{
int x,i;
double S=1;
printf("Please input x : ");
scanf("%d", &x); fflush(stdin);
for(i=1; i<=x; i++)
{
S += pow(x,i)/ factorial(i);
}
printf("%lf", S);
getch();
}
```

Output



```
Please input x : 3
13.000000
```