C CONCEPTS

1.1 Introduction

C is a remarkable language. Designed originally by **Dennis Ritchie**, working at **AT&T Bell** Laboratories in New Jersey, it has increased in use until now it may well be one of the most widely-written computer languages in the world. C is a structured language. It allows variety of programs in small modules. It is easy for debugging, testing, and maintenance if a language is a structured one.

1.2 Structure of a C program:

```
Include header file section
Global declaration section
Main()
{
Declaration part
Executable part
}
User-defined functions
{
Statements
}
```

Include header file section:

C program depends upon some header files for function definition that are used in program. Each header file by default is extended with .h. The header file should be included using # include directive as given here.

Global declaration:

This section declares some variables that are used in more than one function. These variables are known as global variables. This section must be declared outside of all the functions.

Function main:

Every program written in C language must contain main () function. The function main() is a starting point of every C program. The execution of the program always begins with the function main ().

Declaration part:

The declaration part declares the entire variables that are used in executable part. The initializations of variables are also done in this section. Initialization means providing initial value to the variables

Executable part:

This part contains the statements following the declaration of the variables. This part contains a set of statements or a single statement. These statements are enclosed between the braces.

User defined function:

The functions defined by the user are called user-defined functions. These functions are generally defined after the main () function.

1.3 Steps for executing the program

1. ***Creation of program:***
   Programs should be written in C editor. The file name does not necessarily include extension C. The default extension is C.
2. ***Compilation of a program:***
   The source program statements should be translated into object programs which is suitable for execution by the computer. The translation is done after correcting each statement. If there is no error, compilation proceeds and translated program are stored in another file with the same file name with extension ".obj".
3. ***Execution of the program:***
   After the compilation the executable object code will be loaded in the computers main memory and the program is executed.

1.4 C Character set:

Letters	Digits	White Spaces
Capital A to Z	All decimal digits 0 to 9	Blank space
Small a to z		Horizontal tab
		Vertical tab
		New line
		Form feed

Special Characters:

,	Comma	&	Ampersand
.	Dot	^	Caret
;	Semicolon	*	Asterisk
:	Colon	-	Minus
„	Apostrophe	+	Plus
“	Quotation mark	<	Less than
!	Exclamation mark	>	Greater than
|	Vertical bar	()	Parenthesis left/right
/	Slash	[ ]	Bracket left/right
\	Back slash	{}	Braces left/right
~	Tilde	%	Percent
_	Underscore	#	Number sign or Hash
$	Dollar	=	Equal to
?	Question mark	@	At the rate

1.5 Delimiters:

Delimiters	Use
: Colon	Useful for label
; Semicolon	Terminates the statement
( ) Parenthesis	Used in expression and function
[ ] Square Bracket	Used for array declaration
{ } Curly Brace	Scope of the statement
# hash	Preprocessor directive
, Comma	Variable separator

1.6 C Keywords:

Auto	Double	Int	Struct
Break	Else	Long	Switch
Case	Enum	Register	Typedef
Char	Extern	Return	Union
Const	Float	Short	Unsigned
Continue	For	Signed	Void
Default	Goto	Sizeof	Volatile
Do	If	Static	while

1.7 Identifiers:

Identifiers are names of variables, functions, and arrays. They are user-defined names, consisting sequence of letters and digits, with the letter as the first character,

1.8 Constants:

Values do not change during the execution of the program

Types:

1. **Numerical constants:**

    - Integer constants

These are the sequence of numbers from 0 to 9 without decimal points or fractional part or any other symbols. It requires minimum two bytes and maximum four bytes.

Eg: 10,20, +30, -14

- Real constants
  It is also known as floating point constants.

  Eg: 2.5, 5.342

2. **Character constants:**

- Single charcter constants
  A charcter constant is a single character. Charcters are also represented with a single digit or a single special symbol or white space enclosed within a pair of single quote marks

  Eg: „a","8", „ ”"

- String constants
  String constants are sequence of charcters enclosed within double quote marks.

  Eg: “Hello”, “india”,”444”

1.9 Variables:

It is a data name used for storing a data value. Its value may be changed during the program execution. The value of variables keeps on changing during the execution of a program.

1.10 Data types:

Data type	**Size (Bytes)**	**Range**	**Format Specifiers**
Char	1	-128 to 127	%c
Unsigned char	1	0 to 255	%c
Short or int	2	-32,768 to 32, 767	%i or %d
Unsigned int	2	0 to 655355	%u
Float	4	3.4e-38 to +3.4e +38	%f or %g
Long	4	-2147483648 to 2147483647	%ld
Unsigned long	4	0 to 4294967295	%lu
Double	8	1.7e-308 to 1.7e+308	%lf
Long double	10	3.4e-4932 to 1.1e+4932	%lf

1.11 Operators:

It indicates an operation to be performed on data that yields value.

Types:

Type of Operator	**Symbolic representation**
Arithmetic operators	+,-,*,/,%
Relational operators	>,<,==,>=,<=,!=
Logical operators	&&, ||, !=
Increment and decrement operator	++ and --
Assignment operator	=
Bitwise operator	&,|,^,>>,<<, ~
Comma operator	,
Conditional operator	?:

1.12 Input and Output:

Reading data from input devices and displaying the results on the screen are the two main tasks of any program.

Formatted functions:

- The formatted input/output functions read and write all types of values

Input	*Output*
Scanf()	printf()

Unformatted functions:

- The unformatted input/output functions only work with the charcter data type

Input	*Output*
getch()	putch()
getche()	putchar()
getchar()	put()
gets()	

1.13 Decision statements:

It checks the given condition and then executes its sub-block. The decision statement decides the statement to be executed after the success or failure of a given condition.

Types:

1. If statement
2. If-else statement
3. Nested if-else statement
4. Break statement
5. Continue statement
6. goto statement
7. switch() statement
8. nested switch ()case
9. switch() case and nested if

Statement	Syntax

If statement	if(condition)
	Statement;

If-else statement	If (condition)
	{
	Statement 1;
	Statement 2;
	}
	else
	{
	Statement 3;
	Statement 4;
	}

Nested if-else statement	If (condition)
	{
	Statement 1;
	Statement 2;
	}
	Else if (condition)
	{
	Statement 3;
	Statement 4;
	}
	Else
	{
	Statement 5;
	Statement 6;
	}

Break statement	Break;
Continue statement	Continue;
Goto statement	goto label;

Switch() statement	Switch (variable or expression)
	{
	Case constant A:
	Statement;
	Break;
	
	Case constant B:
	Statement;

	Break;
	
	Default:
	Statement;
	}

1.14 Loop Control statements:

Loop is a block of statements which are repeatedly executed for certain number of times.

Types:

1. for loop
2. nested for loops
3. while loop
4. do while loop
5. do-while statement with while loop

Statement	**Syntax**

For loop	For(initialize counter; test condition; re-evaluation parameter
	{
	Statement;
	Statement;
	}

Nested for loop	for(initialize counter; test condition; re-evaluation parameter)
	{
	Statement;
	Statement;
	for(initialize counter; test condition; re-evaluation parameter)
	Statement;
	Statement;
	}
	}

While loop	While (test condition)
	{
	Body of the loop
	}

Do while loop	do
	{
	Statement;
	}
	While(condition);

Do-while with while loop	Do while(condition)
	{
	Statement;
	}
	While (condition);

1.15 Arrays:

It is a collection of similar data types in which each element is located in separate memory locations.

Types:

1. One dimensional array
2. Two dimensional arrays
3. Three or multi dimensional arrays

Operations:

1. Insertion
2. Deletion
3. Searching
4. Sorting
5. Merging

sscanf():

This function allows reading characters from a character array and writes them to another array. This function is similar to scanf(), but instead of reading from standard input it reads data from an array.

sprintf():

This function is similar to the printf() function except for a small difference between them. The printf() function sends the output to the screen whereas the sprint() function writes the values of any data type to an array of characters.

1.16 Strings:

Character arrays are called strings. Group of characters, digits, symbols enclosed within quotation marks are called as strings.

String standard functions:

Functions	Description
Strlen()	Determines the length of a string
Strcpy()	Copies a string from source to destination
Strncpy()	Copies characters of a string to another string up to the specified length
Stricmp()	Compares characters of two strings
Strcmp()	Compares characters of two strings up to the specified length
Strncmp()	Compares characters of two strings up to the specified length
Strnicmp()	Compares characters of two strings up to the specified length
Strlwr()	Converts uppercase characters of a string to lower case
Strupr()	Converts lowercase characters of a string to upper case
Strdup()	Duplicates a string
Strchr()	Determines the first occurrence of a given character in a string
Strrchr()	Determines the last occurrence of a given character in a string
Strstr()	Determines the first occurrence of a given string in another string
Strcat()	Appends source string to destination string
Strrev()	Reverses all characters of a string
Strset()	Sets all characters of a string with a given argument or symbol
Strspn()	Finds up to what length two strings are identical
Strpbrk()	Searches the first occurrence of the character in a given string and then displays the string starting from that character

1.17 Functions:

It is a self-contained block or a sub program of one or more statements that performs a special task

Declaration of functions:

```
Function_name (argument/parameter)
Argument declaration;
{
Local variable declaration;
Statement1;
Statement 2;
Return (value);
}
```

Call by value:
In this type, value of actual arguments is passed to the formal arguments and the operation is done on the formal arguments. Any change made in the formal argument does not affect the actual arguments because formal arguments are photo copies of actual arguments.

Call by reference:
In this type, instead of passing values, addresses are passed. Function operates on address rather than values. Here the formal arguments are pointers to the actual argument.

1.18 Recursion:

A function is called repetitively by itself.

1.19 Pointers

A pointer is a memory variable that stores a memory address. It can have any name that is legal for another variable and it is declared in the same fashion like other variables but it is always denoted by „*" operator.

Void pointers:

Pointers can also be declared as a void type. Void pointers cannot be dereferencing without explicit type conversion.

1.20. Structure:

A structure is a collection of one or more variables of different data types grouped together under a single name.

typedef:

By using typedef we can create new data type. The statement typedef is to be used while defining the new data type. The syntax is

typedef type dataname;

type is the data type; dataname is the user-defined name for that type.

Bit-fields:

A bit field provides the exact amount of bits required for storage of values.

Enumerated data type:

Enum is a keyword. It is used for declaring enumeration types. The programmer can create his/her own data type and define what values the variables of these data types can hold.

Eg.

enum month{Jan, Feb, Mar, Apr, may, Jun, Jul, Aug, Sep, Oct, Nov, Dec};

1.21 Union:

Union is a variable, which is similar to the structure. It contains a number of members like structure but it holds only one object at a time.

1.22 Files

File:

File is a set of records that can be accessed through a set of library functions.

File types:

1. Sequential file
2. Random access file

Steps for file operations:

- Opening a file

- Reading or writing a file
- Closing a file

File functions:

Function	Operation
fopen()	Creates a new file for read/write operation
fclose()	Closes a file associated with file pointer
closeall()	Closes all opened files with fopen()
fgetc()	Reads the character from current pointer position and advances the pointer to next character
getc()	Same as fgetc()
fprintf()	Writes all types of data values to the file
fscanf()	Reads all types of data values from a file
putc()	Writes character one by one to a file
fputc()	Same as putc()
gets()	Reads string from the file
puts()	Writes string to the file
putw()	Writes an integer to the file
getw()	Reads an integer from the file
fread()	Reads structured data written by fwrite() function
fwrite()	Writes block of structured data to the file
fseek()	Sets the pointer position anywhere in the file
feof()	Detects the end of the file
ferror()	Reports error occurred while read/write operations
perror()	Prints compilers error messages along with user-defined messages
ftell()	Returns the current pointer position
rewind()	Sets the record pointer at the beginning of the file
unlink()	Removes the specified file from the disk
rename()	Changes the name of the file

Text Modes:

1. W(write):
   This mode opens a new file on the disk for writing. If the file already exists, it will be overwritten without confirmation.
   Syntax:
   fp=fopen("data.txt", "w");

2. r(read):
   This mode opens a pre-existing file for reading. If the file does not exist, then the compiler returns NULL to the pointer.
   Syntax:
   fp=fopen("data.txt", "r");

3. a(append):
   This mode opens a pre-existing file for appending data. If the file does not exist, then the new file is opened, that is, if the file does not exist then the model of "a" is same as "w".
   Syntax:
   fp=fopen("data.txt", "a");

4. w+(write+read)
   It searches for file, if found its contents are destroyed. If the file is not found a new file is created. Returns NULL if fails to open the file. In this mode file can be written and read.
   Syntax:
   fp=fopen("data.txt", "w+");

5. a+(append+read)
   In this mode file can be read and records can be added at the end of file.
   Syntax:
   fp=fopen("data.txt", "a+");

6. r+(read+write):
   This mode is used for both reading and writing. We can both read and write the record in the file. If the file does not exist, then the compiler returns NULL to the pointer.
   Syntax:
   fp=fopen("data.txt", "r+");

Binary modes:

1. wb(write) This mode opens a binary file in write mode
2. rb(read) This mode opens a binary file in read mode
3. ab(append) This mode opens a binary file in append mode, i.e., data can be added at the end of file.
4. r+b(read+write) This mode opens a pre-existing file in read and write mode
5. w+b(read+write) This mode creates a new file in read and write mode
6. a+b(append+write) This mode opens a file in append mode, i.e. data can be written at the end of the file

1.23 Command line arguments

Command:

An executable program that performs a specific task for operating system is called as command.

Command line arguments:

Arguments are associated with the commands; hence these arguments are called as command line arguments.

Application of Command line arguments:

1. Type 2.Del 3. Rename

Environment variables:

Environment variable provide different settings/ path related to operating system.

C PROGRAMS

1. Write a c program to find out the Greatest of Three Numbers

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
printf("enter the three numbers\n");
scanf("%d%d%d",&a,&b,&c);
printf("the greatest number is\n");
if((a>b)&&(a>c))
printf("a is greatest %d",a);
else if(b>c)
printf("b is greatest %d",b);
else
printf("c is greatest %d",c);
getch();
}
```

Output:

```
enter the three numbers
90
89
67
a is greatest 90
```

2. Write a C program to find out the Area and Circumference of Circle

```
#include<stdio.h>
#include<conio.h>
void main()
{
float a,b,c;
clrscr();
printf("enter the r value\n");
scanf("%f",&r);
a=3.14*r*r;
b=2*3.14*r;
printf("area=%f\n",a);
printf("circumference=%f\n",b);
getch();
}
```

Output:

```
enter the r value
3 area=28.26
circumference=18.84
```

3. Write a C program to find out the Average of three Real Numbers

```
#include<stdio.h>
#include<conio.h>
void main()
{
float a,b,c,x;
clrcsr();
printf("enter the three real numbers:\n");
scanf("%f%f%f",&a,&b,&c);
x=(a+b+c)/3;
printf("average=%f\n",x);
getch();
}
```

Output:

```
enter the three real numbers:
3
4
5
average=4.000000
```

4. Write a C program to find out the Sum of two Numbers

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
printf(“enter the two numbers:\n”);
scanf(“%d%d”,&a,&b);
c=a+b;
printf(“sum=%d”,c);
getch();
}
```

Output:

```
enter the two numbers:
5
4
sum=9
```

5. Write a C program to convert Hour into Minutes

```
#include<stdio.h>
#include<conio.h>
void main()
{
float h,m;
clrscr();
printf(“enter the hour:\n”);
scanf(“%f,&h); m=n*60;
printf(“minutes=%f”,m);
getch();
}
```

Output:

```
enter the hour:
8
minutes=480.000000
```

6. Write a C program to find out the Simple Interest

```
#include<stdio.h>
#include<conio.h>
void main()
{
float p,n,r,s;
clrscr();
printf(“enter the p,n,r value:\n”);
scanf(“%f%f%f”,&p,&n,&r);
s=(p*n*r)/100;
printf(“simple interest=%f”,s);
getch();
}
```

Output:

```
enter the p,n,r value:
30000
2
3
simple interest=1800.000000
```

7. Write a C program to convert Celsius to Fahrenheit

```
#include<stdio.h>
#include<conio.h>
void main()
{
float f,c;
clrscr();
printf("enter the celsius value:\n");
scanf("%f",&c); f=((c*9)/5)+32;
printf("fahrenheit value=%f",f);
getch();
}
```

Output:

```
enter the celsius value:
37
fahrenheit value=98.599998
```

8. Write a C program to find out the Area and Perimeter of Rectangle

```
#include<stdio.h>
#include<conio.h>
void main()
{
float a,b,l,p;
clrscr();
printf("enter the l and b value:\n");
scanf("%f%f",&l,&b);
a=l*b;
p=2*(l+b);
printf("area=%f\n perimeter=%f\n",a,p);
getch();
}
```

Output:

```
enter the l and b value:
6
8 area=48.000000
perimeter=28.000000
```

9. Write a C program to find out the Area and Perimeter of Square

```
#include<stdio.h>
#include<conio.h>
void main()
{
float s,a,p;
clrscr();
printf("enter the s value:\n");
scanf("%f",&s);
a=s*s;
p=4*s;
printf("area=%f\n perimeter=%f\n",a,p);
getch();
}
```

Output:

```
enter the s value:
5 area=25.000000
perimeter=20.000000
```

10. Write a C program to find out the Sum and Percentage of five Marks

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

```
{
int a,b,c,d,e,s;
float x;
clrscr();
printf("enter the 5 marks:\n");
scanf("%d%d%d%d%d",&a,&b,&c,&d,&e")
s=a+b+c+d+e;
x=s/5;
printf("sum=%d \npercentage=%f\n",s,x);
getch();
}
```

Output:

```
enter the 5 marks:
87
98
78
76
89 sum=428
percentage=85.000000
```

11. Write a C program for Swapping two Values without Using Temporary Variables

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
clrscr();
printf("enter the two values\n");
scanf("%d%d",&a,&b);
a=a+b;
b=a-b;
a=a-b;
printf("a=%d\nb=%d\n",a,b);
getch();
}
```

Output:

```
enter the two values:
9
8
a=8
b=9
```

12. Write a C program for Swapping two values Using Temporary Variable

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
printf("enter the two values\n");
scanf("%d%d",&a,&b);
c=a;
a=b;
b=c;
printf("a=%d\nb=%d",a,b);
getch();
}
```

Output:

```
enter the two values:
9
```

```
a=8
b=9
```

13. Write a C program to check the given year is Leap Year or not

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
clrscr();
printf(“enter the year\n”);
scanf(“%d”,&a);
if(a%4==0)
printf(“leap year”);
else
printf(“not a leap year”);
getch();
}
```

Output:

```
enter the year:
1998
not a leap year.
```

14. Write a C program to check whether the person is eligible to Vote or Not

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
clrscr();
printf(“enter the age\n”);
scanf(“%d”,&a);
if(a>=18)
printf(“eligible to vote”);
else
printf(“not eligible”);
getch();
}
```

Output:

```
enter the age:
21
eligible to vote
```

15. Write a C program to find out the given number is Greater than100 or Not

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
clrscr();
printf(“enter the number\n”);
scanf(“%d”,&a);
if(a>100)
printf(“greater than 100”);
else
printf(“less than 100”);
getch();
}
```

Output:

```
enter the number
366
```

greater than 100

16. Write a C program to find out the biggest of two Numbers:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b;
clrscr();
printf("enter the 2 numbers");
scanf("%d%d",&a,&b);
if(a>b)
printf("a is biggest");
else
printf("b is biggest");
getch();
}
```

Output:

```
enter the two numbers:
6
7
b is biggest.
```

17. Write a C program to find out the given number is Odd or Even Number

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
clrscr();
printf("enter the number\n");
scanf("%d",&a);
if(a%2==0)
printf("even number");
else
printf("odd number");
getch();
}
```

Output:

```
enter the number
9
odd number
```

18. Write a C program to convert Fahrenheit to Celsius

```
#include<stdio.h>
#include<conio.h>
void main()
{
float f,c;
clrscr();
printf("enter the fahrenheit value\n");
scanf("%f",&f); c=((f-
32)*5)/9; printf("celsius
value=%f",f); getch();
}
```

Output:

```
enter the fahrenheit value:
98.6
celsius value=37
```

19. Write a C program to find out the Greatest of two Numbers Using Conditional Operator

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

```
#include<conio.h>
void main()
{
int a,b;
clrscr();
printf("enter the 2 numbers");
scanf("%d%d",&a,&b);
(a>b?printf("a is greater"):printf("b is greater"));
getch();
}
```

Output:

```
enter the two numbers
6
3
a is greater
```

20. Write a C program to find out the Roots of Quadratic Equation

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
int a,b,c,d;
float x1,x2;
clrscr();
printf("\nenter the values of a,b,c\n");
scanf("%d%d%d",&a,&b,&c);
d=b*b-4*a*c;
if(d==0)
{
printf("\nthe roots are real and equal");
x1=-b/(2*a);
printf("%f\n",x1);
}
else if(d<0)
{
printf("\nthe roots are imaginary\n");
x1=-b/(2*a); x2=sqrt(-
d)/(2*a);
printf("%2f+i%2f\n",x1,x2);
printf("%2f-i%2f\n",x1,x2);
}
else
{
printf("\n the roots are real and distinct");
x1=-b+sqrt(d)/(2*a); x2=-
b-sqrt(d)/(2*a);
printf("\n%f\n%f",x1,x2);
}
getch();
}
```

Output:

```
enter the values of a,b,c
2
8
3
the roots are real and distinct
-6.418861
 -9.581139
```

21. Write a C program to perform Menu Driven Calculator

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

```
{
int a,b,c,ch; clrscr();
printf("\n1.add\n2.subtract\n3.multiply\n4.division\n5.remainder\n);
printf("\nenter your choice\n");
scanf("%d",&ch);
switch(ch)
{
case1:
printf("\nenter values of a and b\n");
scanf("%d%d",&a,&b);
c=a+b;
printf("\nthe answer is %d",c);
break;
case2:
printf("\nenter values of a and b\n");
scanf("%d%d",&a,&b);
c=a-b;
printf("\nthe answer is %d",c);
break;
case3:
printf("\nenter values of a and b\n");
scanf("%d%d",&a,&b);
c=a*b;
printf("\nthe answer is %d",c);
break;
case4:
printf("\nenter values of a and b\n");
scanf("%d%d",&a,&b);
c=a/b;
printf("\nthe answer is %d",c);
break;
case5:
printf("\nenter values of a and b\n");
scanf("%d%d",&a,&b);
c=a%b;
printf("\nthe answer is %d",c);
break;
default:
printf("\nenter the correct choice");
break;
}
getch();
}
```

Output:

```
1.add
2.subtract
3.multiply
4.division
5.remainder
enter your choice
2
enter the values  of a and b
7
4
the answer is 3
```

22. Write a C program to covert Decimal to Binary Conversion

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
int no,r,sum=0,i=0;
clrscr();
printf("\nenter the number\n");
scanf("%d",&no);
while(no>0)
{ r=no%2;
sum=sum+pow(10,i)*r;
no=no/2;
i++;
}
printf("\nthe binary value is %d",sum);
getch();
}
```

Output:

```
enter the number
8
the binary value is 1000
```

23. Write a C program to display the Number and its Square

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,i;
clrscr();
printf("enter the value of n\n");
scanf("%d",&n);
for(i=1;i<=n;i++)
printf("the number is %d and its square is %d\n",i,i*i);
getch();
}
```

Output:

```
enter the value of n
3
the number is 1 and its square is 1
the number is 2 and its square is 4
the number is 3 and its square is 9
```

24. Write a C program to find out the Sum and Average of First N Numbers

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

Output:

```
enter the value of n
5 sum=15
average=3
```

25. Write a C program to Print the prime numbers From N to 1

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n;
clrscr();
printf("\nenter the value of n\n");
scanf("%d",&n);
for( ;p>0;p--)
printf("%d\n",p);
getch();
}
```

Output:

```
enter the value of n
4
4
3
2
1
```

26. Write a C program to find out the Sum of N Numbers

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,sum=0;
clrscr();
printf("\nenter the value of n\n");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
sum=sum+i;
}
printf("sum=%d\n",sum);
getch();
}
```

Output:

```
enter the value of n
10
sum=55
```

27. Write a C program to find out the Fibonacci Series

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a=0,b=1,c=0,i,n;
clrscr();
printf("enter the number of terms\n");
printf("%d\n%d",a,b);
for(i=3;i<=n;i++)
{
c=a+b;
printf("%d\n",c);
a=b;
b=c;
}
getch();
}
```

Output:

```
enter the number of terms
```

```
4
0
1
1
2
```

28. Write a C program to find out the Factorial of a Given Number

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,i,fact=1;
clrscr();
printf(“enter the  number\n”);
scanf(“%d”,&n);
for(i=n;i>=1;i--)
{
fact=fact*i;
}
printf(“the factorial of given number is %d”,fact);
getch();
}
```

Output:

```
enter the number
5
the factorial of given number is 120
```

29. Write a C program to find out the Sum of N Numbers Using While Loop

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i=1,sum=0;
clrscr();
printf(“enter the  value of n\n”);
scanf(“%d”,&n);
while(i<=n)
{
sum=sum+i;
i++;
}
printf(“sum=%d”,sum);
getch();
}
```

Output:

```
enter the value of n
5
sum=15
```

30. Write a C program to calculate the Electric Energy Bill

```
#include<stdio.h>
#include<conio.h>
void main()
{
float r,a=2.5,b=3.5,c=1.5;
clrscr();
printf(“enter the readings\n”);
scanf(“%f”,&r); if(r>=200)
printf(“rupees=%f”,r*b);
else if((r>=100)&&(r<200))
printf(“rupees=%f”,r*a);
else
printf("rupees=%f",r*c);
getch();
}
```

Output:

```
enter the readings
140
rupees=350
```

31. Write a C program to display the Prime Numbers between 100 and 500

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j;
clrscr();
printf("the prime numbers are:\n");
for(i=100;i<=500;i++)
{
for(j=2;j<i;j++)
{
if(i%j==0)
break;
} if(i==j)
printf("%d\t",i);
}
getch();
}
```

Output:

```
the prime numbers are:
101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223 227 229 233 239
241 251 257 263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 367 373 379 383 389 397 401 409
419 421 431 433 439 443 449 457 461 463 467 479 487 491 499
```

32. Write a C program to find out the given number is Armstrong Number or not

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
int n,k,r,sum=0;
clrscr();
printf("enter the number\n");
scanf("%d",&n);
k=n;
while(n!=0)
{ r=n%10;
sum=sum+pow(r,3);
n=n/10;
}
if(sum==k)
printf("the number is armstrong");
else
printf("the number is not armstrong");
getch();
}
```

Output:

```
enter the number
153
the number is armstrong
```

33. Write a C program to find out the given number is Palindrome Number or Not

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,k,r,sum=0;
clrscr();
printf(“enter the number\n”);
scanf(“%d”,n);
k=n;
while(n!=0)
{ r=n%10;
sum=sum*10+r;
n=n/10;
}
if(sum==k)
printf(“the number is palindrome”);
else
printf(“the number is not palindrome”);
getch();
}
```

Output:

```
enter the number
323
the number is palindrome
```

34. Write a C program to find out the Maximum Value in the Array

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[5],max,i;
clrscr();
printf(“enter elements for the array\n”);
for(i=0;i<5;i++)
scanf(“%d”,&a[i]);
max=a[0];
for(i=1;i<5;i++)
{
if(max<a[i])
max=a[i];
}
printf(“the maximum value is%d”,max);
getch();
}
```

Output:

```
enter the elements for array
4
6
3
8
5
the maximum value is 8
```

35. Write a C program to perform Matrix Multiplication

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[10][10],b[10][10],c[10][10],r1,r2,c1,c2,i,j,k;
clrscr();
printf(“enter the no.of rows and columns for 1st matrix:\n”);
scanf(“%d%d”,&r1,&c1);
printf("enter the values of 1st  matrix:\n");
for(i=0;i<r1;i++)
{ for(j=0;j<c1;j++)
scanf("%d",&a[i][j]);
}
printf("enter the no.of rows and columns for 2nd  matrix:\n");
scanf("%d%d",&r2,&c2);
printf("enter the values of 2nd matrix:\n");
for(i=0;i<r2;i++)
{ for(j=0;j<c2;j++)
scanf("%d",&b[i][j]);
}
if(c1==r2)
{
for(i=0;i<r1;i++)
{
 for(j=0;j<c2;j++)
{ c[i][j]=0;
for(k=0;k<c1;k++)
{
c[i][j]=c[i][j]+a[i][k]+b[k][j];
}
}
}
printf("resultant matrix is:\n");
for(i=0;i<r1;i++)
{
for(j=0;j<c1;j++)
{
printf("%d\t",c[i][j]);
}
printf("\n");
}
}
else
printf("the matrix are not multiplied");
getch();
}
```

Output:

```
enter the no.of rows and columns for 1st  matrix:
3
3
enter the values of 1st  matrix:
1
1
1
1
1
1
1
1
1
enter the rows and columns for 2nd  matrix:
3
3
enter the values of 2nd  matrix:
1
1
1
1
1
1
```

```
1
1
1
resultant matrix is:
3  3  3
3  3  3
3  3  3
```

36. Write a C program to display the following output

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

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,k; clrscr();
for(i=1;i<=3;i++)
{
for(j=3;j>=i;j--)
{
printf(“  “);
}
for(k=1;k<=i*2-1;k++)
{
printf(“*”);
}
printf(“\n”);
}
getch();
}
```

37. Write a C program to display the following Output

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

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,k; clrscr();
for(i=1;i<=5;i++)
{
for(j=5;j>i;j--)
printf(“  “);
for(k=1;k<=i;k++)
printf(“*”);
printf(“\n”);
}
getch();
}
```

38. Write a C program to perform searching the element in an Array

```
#include<stdio.h>
#include<conio.h>
void main()
{
int j=0,n,x[5];
clrscr();
printf(“enter the elements of array\n”);
for(j=0;j<5;j++)
scanf(“%d”,&x[j]);
printf(“enter the element to search\n”);
scanf(“%d”,&n);
for(j=0;j<5;j++)
{
if(x[j]==n)
break;
} if(x[j]==n)
printf(“element found”);
else
printf(“element not found”);
getch();
}
```

Output:

```
enter the elements of array:
1
2
3
4
5
enter the elements to search
3
element found
```

39. Write a C program to perform inserting elements in an Array

```
#include<stdio.h>
#include<conio.h>
void main()
{
int num[20],j,p,n,s;
clrscr();
printf(“enter the number of elements\n”);
scanf(“%d”,&n);
printf(“enter the elements of array\n”);
for(j=0;j<n;j++)
scanf(“%d”,&num[j]);
printf(“enter the element and positon to be inserted\n”);
scanf(“%d%d”,&s,&p);
p--;
for(j=n;j!=p;j--)
{
num[j]=num[j-1];
} num[j]=s;
for(j=0;j<=n;j++)
printf(“%d”,num[j]);
getch();
}
```

Output:

```
enter the number of elements
4
enter elements
1
2
3
5
enter the element and position to be inserted
4
4
1  2  3  4
```

40. Write a C program to perform deleting element in an Array

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

```
void main()
{
int num[20],j,p,n,s;
clrscr();
printf("enter the number of elements\n");
scanf("%d",&n);
printf("enter the elements of array\n");
for(j=0;j<n;j++)
scanf("%d",&num[j]);
printf("enter the position to delete\n");
scanf("%d",&p);
p--;
for(j=p;j<n;j++)
num[j]=num[j+1];
for(j=0;j<n-1;j++)
printf("%d",num[j]);
getch();
}
```

Output:

```
enter the number of elements
3
enter elements
1
2
3
enter the position to delete
2
1  3
```

41. Write a C program to perform sorting the elements in an array

```
#include<stdio.h>
#include<conio.h>
void main()
{
int num[20],j,k,n,s=0;
clrscr();
printf("enter the number of elements\n");
scanf("%d",&n);
printf("enter the elements of array\n");
for(j=0;j<n;j++)
{
scanf("%d",&num[j]);
s=s+num[j];
}
for(k=0;k<s;k++)
{
for(j=0;j<n;j++)
{ if(num[j]==k)
printf("%d",num[j]);
}
}
getch();
}
```

Output:

```
enter the number of elements
3
5
3
8
3  5  8
```

42 Write a C program to perform merging the elements in an Array

```
#include<stdio.h>
#include<conio.h>
void main()
{
int j,h=0,k=0;
int  x[4]={1,2,3,4};
int  y[4]={5,6,7,8};
int z[8];
clrscr();
printf("array x:\n");
for(j=0;j<4;j++)
printf("%d",x[j]);
printf("array y:\n");
for(j=0;j<4;j++)
printf("%d",y[j]);
j=0;
while(j<8)
{ if(j%2==0)
z[j]=x[k++];
else
z[j]=y[h++];
j++;
}
printf("array z:\n");
for(j=0;j<8;j++)
printf("%d",z[j]);
getch();
}
```

Output:

```
array x:
1  2  3  4
array y:
5 6 7 8
array z:
1 2 3 4 5 6 7 8
```

43 Write a C program to perform the sorting the given Strings in Ascending Order

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
int i,j,n,x;
char str[20][20],str1[20][20];
clrscr();
printf("enter the number of strings:\n");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("\nenter str[%d]",i+1);
scanf("%s",&str[i]);
}
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
x=strcmp(str[i],str[j])
if(x>0)
{
strcpy(str[1],str[j]);
strcpy(str[j],str[i]);
strcpy(str[i],str[1]);
}
}
}
printf("\nthe sorted strings in ascending order is\n");
for(i=0;i<n;i++)
{
printf("\n%s",str[i]);
}
getch();
}
```

Output:

```
enter the number of strings:
3
enter str[1] raja
enter str[2] vignesh
enter str[3] adhi
the sorted strings in ascending order is
adhi
raja
vignesh
```

44. Write a C program to find out the Fibonacci Series using Recursive Function

```
#include<stdio.h>
#include<conio.h>
int fibo(int,int)
int t1,t2,t3,count;
void main()
{
printf("enter the number of terms\n");
scanf("%d",&n);
t1=0; t2=1;
printf("%d\t%d",t1,t2);
count=2;
fibo(t1,t2);
getch();
}
int fibo(int t1,int t2)
{
if(count>=n)
return 0;
else
{ t3=t1+t2;
printf("\t%d",t3);
count++;
t1=t2;
t2=t3;
fibo(t1,t2);
}
}
```

output:

```
enter the number of terms
5
0 1 1 2 3
```

45. Write a C program to find out the Swapping of two Values using Functions

```
#include<stdio.h>
#include<conio.h>
int swapval(int,int);
int swapref(int*,int*);
int a,b;
void main()
{
clrscr();
printf("enter the two values\n");
scanf("%d%d",&a,&b);
printf(“pass by value\n”);
printf(“before function call a=%d b=%d “,a,b);
swapval(a,b);
printf(“after function swapval a=%d b=%d “,a,b);
printf(“pass by reference\n”);
printf(“before function call a=%d b=%d “,a,b);
swapref(&a,&b);
printf(“after function swapref a=%d b=%d “,a,b);
getch();
}
swapval(int x,int y)
{
int  t;
t=x;
x=y;
y=t;
printf(“\nwith swap val x=%d y=%d”,x,y);
}
swapref(int*x,int*y)
{
int *t;
*t=*x;
*x=*y;
*y=*t;
printf(“\nwith swapref x=%d y=%d “,*x,*y);
}
```

Output:

```
give two numbers
5
6
pass by value
 before function call a=5 b=6
with swapval x=6 y=5
after function swapval a=5 b=6
pass by reference
before function call a=5 b=6
with swapref x=6 y=5l
after function swapref a=6 b=5
```

46. Write a C program to perform the Substring Replacement

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char str[50],str1[15],str2[15],temp[50];
char *ptr;
int cnt;
clrscr();
printf(“enter a line of text..... \n”);
gets(str);
printf(“enter the string to be replaced... \n”);
gets(str1);
printf(“enter the replacing string...”);
gets(str2);
printf(“\n the replaced line of the text....”);
while(1)
{
ptr=strstr(str,str1);
if(ptr==“\o“)
break; cnt=ptr-str;
strncpy(temp,str,cnt);
temp[cnt]=“10“;
strcat(temp,str+cnt+strlen(srt1));
strcpy(str1,temp);
puts(str);
}
getch();
}
```

Output:

```
enter the line of text... i love india
enter the string to be replaced..india
enter the replacing string...my parents
the replaced line of text
i love my parents
```

47. Write a C program to transpose of a Matrix using Function

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
void trans(int,int,int[10][10]);
int i,j,a,b,m[10][10];
clrcsr();
printf("enter the rows and columns of matrix");
scanf("%d%d",&a,&b);
printf("enter the elements");
for(i=1;i<=a;i++)
{
for(j=1;j<=b;j++)
{
printf("enter m[%d][%d]..."i,i);
scanf("%d",&m[i][j]);
}
}
printf("\n before transpose\);
for(i=1;i<=a;i++)
{
for(j=1;j<=b;j++)
{
printf("%d\t",m[i][j]);
}
printf("\n");
}
trans(a,b,m);
getch();
}
void trans(inta,intb,intm[10][10])
{
int i,j;
printf("after transpose");
for(j=1;j<=b;j++)
{
for(i=1;i<=a;i++)
{
printf("\t%d",m[i][j]);
}
printf("\n");
}
}
```

Output:

```
enter the rows and columns of matrix..3
3
enter the elements
enter m[1][1]=1
enter m[1][2]=2
enter m[1][3]=3
enter  m[2][1]=4
enter  m[2][2]=5
enter  m[2][3]=6
enter  m[3][1]=7
enter  m[3][2]=8
enter m[3][3]=9
before transpose 1 2 3
4 5 6
7 8 9
after transpose 1 4 7
2 5 8
3 6 9
```

48. Write a C program to find out the Standard Deviation using Function

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
float mean(int a[],int n);
float std(int a[],int n,float m);
void main()
{
float m,sd;
int n,a[10],i;
clrscr();
printf(“\nenter the number of values\n”);
scanf(“%d”,&n);
printf(“\n enter the elements\n”);
for(i=0;i<n;i++)
scanf(“%d”,&n);
m=mean(a,n);
printf(“mean=%f\n”,m);
sd=std(a,n,m);
printf(“\n sd=%f”,sd);
getch();
}
flaot mean (inta[],intn)
{
float f;
int sum=0;
for(i=0;i<n;i++)
sum=sum+ a[i];
f=(float)sum/n;
return f;
}
float std(int a[],int n,float m)
{
int i;
float std,sum=0.0,d;
for(i=0;i<n;i++)
{
d=a[i]-m;
a=d*d;
sum=sum+d;
}
sd=sqrt(sum\n);
return sd;
}
```

Output:

```
enter the number of values
5
enter the elements
2
4
6
8
```

```
10
mean=6.000000
sd=2.828427
```

49. Write a C program to find out the Palindrome without using String Function

```
#include<stdio.h>
#include<conio.h>
void main()
{
char str[20];
int i,j,flag=0;
clrscr();
printf(“enter a string:\n”);
scanf(“%s”,str);
i=o;
while(str[i]!=‵‵10‶‶)
i++;
j=j-1;
for(i=0;i<=j;i++,j--)
{
if(str[i]!=str[j])
{
flag=1
break;
}
}
if flag==0
printf(“it is palindrome”);
else
printf(“it is not a palindrome”);
getch();
}
```

Output:

```
enter a string:
malayalam
it is palindrome
```

50. Write a C program to perform Mark list Analysis using Structures

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
struct stud
{
int rno;
char name[15];
int marks[5];
int total;
float avg;
char class[15];
}
st[10],temp;
int i,n,j;
clrscr();
printf(“\n enter\n”);
scanf(“%d”,&n);
for(i=1;i<=n;i++)
{
printf(“\n enter the roll no..”);
scanf(“%d”,&st[i].rno);
printf(“name...\n”);
scanf(“%s”,&st[i].name);
printf(“enter three marks..”);
for(j=1;j<=3;j++)
scanf(“%d”,&st[i].marks[j]);
}
for(i=1;i<=n;i++)
{
st[i].total=0;
for(j=1;j<=3;j++)
{
st[i].total=st[i].total+st[i].marks[j];
} st[i].avg=st[i].total/30;
if(st[i].avg>=75)
strcpy(st[i].grade,”distinction”);
elseif(st[i].avg>=60)
strcpy(st[i].grade,”first”);
elseif(st[i].avg>=50)
strcpy(st[i].grade,”second”);
else
strcpy(st[i].grade,”fail”);
}
for(i=1;i<=n;i++)
{
for(j=j+1;j<=n;j++)
{
if(st[i].total<st[j].total)
{
temp=st[i];
st[i]=st[j];
st[j]=temp;
}
}
}
printf(“\n the student details in rankwise\n”);
for(i=1;i<=n;i++)
{
printf(“\n\n roll no:%d”,st[i].rno);
printf(“\n name :%s”,st[i].name);
printf(“\n marks in three subjects”);
for(j=1;j<=3;j++)
{
printf(“\n %d,st[i].marks[j]);
}
printf (“\n total: %d”, st[i].total);
printf(“\n  average:%f”,st[i].avg);
printf(“\n grade:%s”,st[i].grade);
}
getch();
}
```

Output:

```
enter
2
enter the roll no...105
name...sheik raja
enter the three marks...89
87
78
enter roll no...110
name...sriram
enter the three marks...98
96
95

the student details in rankwise

roll no:105
name:sheik raja
```

```
marks in three subjects
89
87
78
total:254
average:84.666664
grade:distinction

roll no:110
name:sriram
marks in three subjects
98
96
95 total:289
average:96.3333336
grade:distinction
```

51. Write a C program to shift the input data by two bits right

```
#include<stdio.h>
#include<conio.h>
void main()
{
int x,y;
clrscr();
printf("Read the integer from keyboard(x):");
scanf("%d", &x);
x>>=2;
y=x;
printf("The right shifted data is =%d", y);
getch();
}
```

Output:

```
Read the integer from keyboard(x): 8
The right shifted data is = 2
```

52. Write a C program to shift the input data by three bits left

```
#include<stdio.h>
#include<conio.h>
void main()
{
int x,y;
clrscr();
printf("Read the integer from keyboard(x):");
scanf("%d", &x);
x<<=3;
y=x;
printf("The right shifted data is =%d", y);
getch();
}
```

Output:

```
Read the integer from keyboard(x): 2
The right shifted data is = 16
```

53. Write a C program to use bitwise AND operator between the two integers and display the results.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
printf("Read the integer from keyboard(a&b):");
scanf("%d%d", &a, &b);
c=a&b;
printf("The answer after AND operation is (c)=%d", c);
getch();
}
```

Output:

```
Read the integer from keyboard(a&b): 8  4
The answer after AND operation is (c)=0
```

54. Write a C program to operate OR operation on two integers and display the result.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
printf("Read the integer from keyboard(a&b):");
scanf("%d%d", &a, &b);
c=a|b;
printf("The answer after OR operation is (c)=%d", c);
getch();
}
```

Output:

```
Read the integer from keyboard (a&b): 8  4
The answer after OR operation is (c)= 12
```

55. Write a C program to operate XOR operation on two integers and display the result.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
printf("Read the integer from keyboard(a&b):");
scanf("%d%d", &a, &b);
c=a^b;
printf("The answer afterX OR operation is (c)=%d", c);
getch();
}
```

Output:

```
Read the integer from keyboard (a&b): 8  4
The answer after XOR operation is (c)= 10
```

56. Write a C program to read and print the integer value using character variable.

```
#include<stdio.h>
#include<conio.h>
void main()
{
Char a;
clrscr();
printf("Enter value of „A‟:");
scanf("%d", &a);
printf("A=%d", a);
getch();
}
```

Output:

```
Enter value of „A‟: 255
A=255
Enter value of „A‟: 256
A=0
```

57. Write a C program to check whether the entered number is less than 10. If yes, display the same.

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

```
#include<conio.h>
void main()
{
int v;
clrscr();
printf("Enter the number:");
scanf("%d", &v);
if(v<10)
printf("\n Number entered is less than 10");
sleep(2);
getch();
}
```

Output:

```
Enter the number:9
Number entered is less than 10
```

58. Write a C program to check equivalence of two numbers.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int m,n;
clrscr();
printf("Enter two numbers:");
scanf("%d%d", &m, &n);
if(m-n= =0)
printf("\n  Two numbers are equal");
getche();
}
```

Output:

```
Enter two numbers: 5 5
Two numbers are equal
```

59. Write a C program to calculate the square of those numbers only whose least significant digit is 5

```
#include<stdio.h>
#include<conio.h>
void main()
{
int s,d;
clrscr();
printf("Enter a number:");
scanf("%d", &s);
d=s%10;
if(d= =5)
{
s=s/10;
printf("\n square=%d%d", s*s++, d*d);
}
Else
Printf("\n Invalid number");
}
```

Output:

```
Enter a number: 25
Square =625
```

60. Write a C program to convert decimal number to hexadecimal number.

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
void main()
{
Int x, y=30, z;
clrscr();
printf("Enter the number:");
scanf("%d", &x);
printf("\n conversion of decimal to hexadecimal number\n");
for(;;)
{
if(x= =0)
exit(1);
z=x%16;
x=x/16;
gotoxy(y--,5);
switch(z)
{
Case 10:
Printf("A");
Break;
Case 11:
Printf("%c", „B‟);
Break;
Case 12:
Printf(%c", „C");
Break;
Case 13:
Printf("D");
Break; Case
14:
Printf("E");
Break;
Case 15:
Printf("F");
Default:
Printf("%d", z);
}
}
getch();
}
```

Output:

```
Enter the number: 31
Conversion of decimal to Hexa decimal number
1F
```

61. Write a C program to count number of 1s, blank spaces and others using nested switch() statements.

```
#include<stdio.h>
#include<conio.h>
void main()
{
Static int x,s,a,z,o;
chartxt[20];
clrscr();
printf("Enter numbers");
gets(txt);
while(txt[x]!=‟\0‟)
{
Switch(txt[x])
{
case‟‟:
s++;
break;
default:
switch(txt[x])
{
case‟1‟:
a++;
break;
case‟0‟:
z++;
break;
default:
o++‟
}
}
X++;
}
Printf("\n total spaces:%d", s);
Printf("\n total 1s:%d", a);
Printf("\n total 0s:%d", z);
Printf("\n others:%d", o);
Printf("\n string length:%d", s+a+z+o);
}
getch();
}
```

Output:

```
Enter numbers:1110022 222
Total spaces   :1
Total 1s        :1
Total 0s        :2
Others          :5
String length :11
```

62. Write a C program to print five numbers starting from one together with their squares.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i; clrscr();
for(i=1;i<=5;i++)
printf("\n  number: %5d its square : %8d", i, i*i);
getch();
}
```

Output:

```
Number: 1 it‟s square: 1
Number: 2 it‟s square: 4
Number: 3 it‟s square: 9
Number: 4 it‟s square: 16
Number: 5 it‟s square: 25
```

63. Write a C program to evaluate the series given in comments
/*x=1/1+104+1/9...1/n2*/
/*y=1/1+1/8+1/27...1/n3*/

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
int i,n;
float x=0, y=0;
clrscr();
printf("enter value of n:");
scanf("%d", &n);
for(i=1;i<=n; i++)
{
x=x+1(1/pow)i,2));
y=y+(1/pow(i,3));
}
Printf("value of x=%f",x);
printf("\n  value fo y=%f",y);
getche();
}
```

Output:

```
Enter value of n:2
Value of x= 1.2500
Value of y=1.12500
```

64. Write a C program to perform multiplication of two integers by using negative sign.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c,d=0;
clrscr();
printf("\n enter two numbers:");
scanf("%d%d", &a, &b);
for(c=1;c<=b;c++)
d=(d)-(a);
printf("Multiplication of  %d * %d:%d",a,b,d);
getch();
}
```

Output:

```
Enter two numbers: 5 5
Multiplication of 5 *5 : 25
```

65. Write a C program to perform multiplication of two integers by using repetitive addition.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
int a,b,c=1,d=0;
clrscr();
printf("\n enter two numbers:");
scanf("%d%d", &a, &b);
for(;;)
d=d+a;
if(c= =b)
goto stop;
c++;
}
Stop:
printf("Multiplication of  %d * %d:%d",a,b,d);
getch();
}
```

Output:

```
Enter two numbers:  8 4
Multiplication of 8 *4 : 32
```

66. Write a C program to simulate a digital clock.

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
void main()
{
int h,m,s; clrscr();
for(h=1;h<=12;h++)
{ clrscr();
for(m=1;m<=59;m++)
{ clrscr();
for(s=1;s<=59;s++)
{
```

```
Gotoxy(8,8); printf(“hh
mm ss”); gotoxy(8,9);
printf(“%d%d%d”, h,m,s);
sleep(1);
}
}
}
getch();
}
```

Output:

```
Hh  mm  ss
1       1     1
```

67. Write a C program to accept a number and find the sum of its individual digits repeatedly till the result is a single digit.

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
void main()
{
int n,s=0;
clrscr();
printf(“\n enter a number:”);
scanf(“%d”, &n);
printf(“\n Sum of digits till a single digit\n %d”, n);
for(;n=0;)
{
s=s+n%10;
n=n/10;
if(n= =0) &&s>9)
{
Printf(“\n %2d”, s);
N=s;
S=0;
}
}
printf(“\n%2d”,s);
getch();
}
```

Output:

```
Enter  a number:4687
Sum of digits till a single digit
4687
25
7
```

68. Write a C program to display the series of numbers as given below.

1
2 1
3 2 1
4 3 2 1

4 3 2 1
3 2 1
2 1
1

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,j,x;
printf(“\n enter value of x:”);
scanf(“%d”, &x);
clrscr();
for(j=1;j<=x;j++)
{
for(i=j;i>=1; i--)
{
Printf(“%3d”, i);
}
Printf(“\n”);
}
Printf(“\n”);

for(j=x;j<=1;j--)
{
for(i=j;i>=1; i--)
{
Printf(“%3d”, i);
}
Printf(“\n”);
}
getch();
}
```

Output:

```
Enter value of x: 4
1
2 1
3 2 1
4 3 2 1

4 3 2 1
3 2 1
2 1
1
```

69. Write a C program to generate the pyramid structure using numerical

```
#include<stdio.h>
#include<conio.h>
void main()
{
int k,i,j,x,p=34;
clrscr();
printf(“Enter a number:”);
scanf(“%d”, &x);
for(j=0;j<=x;j++)
{ gotoxy(p,j+1);
for(i=0-j;i<=j;i++)
printf(“%3d”, abs(i));
p=p-3;
}
getch();
}
```

Output:

```
 Enter a number: 3

    0
   101
  21012
3 210123
```

70. Write a C program to display array elements in reverse order

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

```
{
int show(int*);
int num[]={12,13,14,15,16,17,18}
clrscr();
show(&num[6]);
}
Show(int*u)
{
Int m=6;
while(m!=-1)
{
printf("\nnum[%d]=%d",m,*u);
u--, m--;
}
return (NULL);
getch();
}
```

Output:

```
Num[6]=18
Num[5]=17
Num[4]=16
Num[3]=15
Num[3]=14
Num[1]=13
Num[0]=12
```

71. Write a C program to calculate triangular number of an entered number with recursion function method.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n,t,tri_num(int);
clrscr();
printf("Enter a number:");
scanf("%d", &n);
t-tri_num(n);
printf("\n Triangular number of %d is %d ", n,t);
}
tri_num(m)
{
int f=0;
if(m= =0)
return(f);
else
f=f+m+tri_num(m-1);
return(f);
getch();
}
```

Output:

```
Enter a number: 5
Triangular number 5 is 15
```

72. Write a C program to accept any string up to 15 characters. Display the elements of string with their element numbers in a separate column.

```
 #include<stdio.h>
#include<conio.h>
#include<process.h>
void main()
{
static char name[15];
int i;
clrscr();
printf("Enter your name:");
gets(name);
printf(“Element no. & character\n”);
for(i=0;i<=15;i++)
{
if(name[i]= =‟\0‟)
exit(1);
printf(“\n %d \t\t%c”, i, name[i]);
}
getch();
}
```

Output:

```
Enter your name:shri

0    s
1    h
2    r
3    i
```

73. Write a C program to print “WELCOME” by using different formats of initialization of array.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char arr1[9]={„W‟,‟E‟,‟L‟,‟‟‟,‟C‟,‟O‟,‟M‟,‟E‟,‟\0};
char arr2[9]=”WELCOME”;
char arr3[9]={{ „W‟},{‟E‟},{‟L}‟,{‟ ‟},{‟C‟},{‟O‟},{‟M‟},{‟E‟}};
clrscr();
printf(“\narray1=%s:”, arr1);
printf(“\narray2=%s:”, arr2);
printf(“\narray3=%s:”, arr3);
getch();
}
```

Output:

```
Array1=WELCOME
Array2=WELCOME
Array3=WELCOME
```

74. Write a C program to count the number of characters in a given string

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char text [20];
int len;
clrscr();
printf(“type text below\n”);
gets(text);
len=strlen(text);
printf(“Length of string=%d”, len);
getch();
}
```

Output:

```
Type text below
Sam
Length of the string=3
```

75. Write a C program to read a name through the keyboard. Determine the length of the string and find its equivalent ASCII codes.

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

```
#include<string.h>
void main()
{
static char name[20];
int i,l;
clrscr();
printf(“enter your name:”);
scanf(“%s”, name);
l=strlen(name);
printf(“your name is %s & ”, name);
printf(“it contains %d characters”, l);
printf(“\n Name & its ASCII Equivalent\n”);
printf(“= = = = = = = = = = = = = = = = \n”);
for(i=0;i<1;i++)
printf(“\n%c\t\t%d”, name[i]name[j]);
getche();
}
```

Output:

```
Enter your name: SACHIN
Your name is SACHIN & it contains 6 characters

Name& its ASCII equivalent
= = = = = = = = = = = = = = = =
S          83
A          65
C          67
H          72
I          73
N          78
```

76. Write a C program to delete all the occurrences of vowels in a given text. Assume that the text length will be of one line.

```
 #include<stdio.h>
#include<conio.h>
void main()
{
 char line[20], line2[80];
int i,j=0;
clrscr();
printf(“enter text below.\n”);
gets(line);
for(i=0;i<80;i++)
{
If(line[i]= =‟a‟ || line[i]= =‟e‟ ||line[i]= =‟i‟ ||line[i]= =‟0‟||line[i]= =‟u‟)
continue;
else
{
line2[j]=line[i];
j++;
}
}
Printf(“\n Text with Vowels: %s”, line);
Printf(“\n Text without vowels:%s”, line2);
getch();
}
```

Output:

```
Enter text below.
anandamurugan
Text with vowels: anandamurugan
Text without vowels: nndmrgn
```

77. Write a C program to to copy contents of one string to another string

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

```
#include<conio.h>
#include<string.h>
void main()
{
Char ori[20], dup[20];
clrscr();
printf(“enter your name:”);
gets(ori);
strcpy(dup,ori);
printf(“original string: %s”, ori);
printf(“\nduplicate string:%s”, dup);
getch();
}
```

Output:

```
Enter your name: SAM
Original string: SAM
Duplicate String: SAM
```

78. Write a C program to convert upper case string to lower case string.

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char upper[15];
clrscr();
printf(“\n Enter a string in upper case:”);
gets(upper);
printf(“After strlwr();%s”, strlwr(upper));
getch();
}
```

Output:

```
Enter a string in upper case: ABCDEFG
After strlwr(): abcdefg
```

79. Write a C program to convert lower case string to upper case string.

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char upper[15];
clrscr();
printf(“\n Enter a string in lower case:”);
gets(upper);
printf(“After strupr();%s”, strupr(upper));
getch();
}
```

Output:

```
Enter a string in lower case: abcdefg
After strupr(): ABCDEFG
```

80. Write a C program to enter the string and get it‟s duplicate

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
Chartext1[20], *text2;
clrscr();
printf(“ Enter text:”);
gets(text1);
text2=strdup(text1);
printf(“original string=%s\n duplicate string=%s”, text1, text2);
getch();
}
```

Output:

```
Enter a text: anandamurugan
Original string: anandamurugan
Duplicate String: anandamurugan
```

81. Write a C program to find first occurrence of a given character in a string

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
Char string[30],ch,*chp;
clrscr();
printf(“ Enter text below”);
gets(string);
printf(“\n character to find:”);
ch=getchar();
chp=strchr(string,ch);
if(chp)
printf(“character %c found in string”,ch);
else
printf(“character %c not found in string”,ch);
getch();
}
```

Output:

```
Enter  text below: anandamurugan
Character to find: r
Character r found in string
```

82. Write a C program to find occurrence of a second string in the first string.

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
Char line[30], line2[30],*chp;
clrscr();
puts(“Enter line1:”);
gets(line1);
puts(“Enter line2:”);
gets(line2);
chp=strstr(line1,line2);
if(chip)
printf(“ ‟%s‟ string is present in given string”, line2);
else
printf(“ ‟%s‟ string is not present in given string”, line2);
getch();
}
```

Output:

```
Enter  line1: anandamurugan is an author
Enter line2: author
„author‟ string is present  in given string
```

83. Write a C program to concatenate two strings without the use of a standard function.

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char name[50],fname[15],sname[15],lname[15];
int i,j,k;
clrscr();
printf(“First name:”);
gets(fname);
printf(“second name:”);
gets(sname);
printf(“last  name:”);
gets(lname);
for(i=0;fname[i]!=‟\0‟;i++)
name[i]=fname[i];
name[i]=‟‟;
for(j=0;sname[j]!=‟\0‟;j++)
name[i+j+1]=sname[j];
name[i+j+1]=‟‟;
for(k=0;lname[k]!=‟\0‟;k++)
name[i+j+k+2]=lname[k];
name[i+j+k+2]=‟\0‟;
printf(“\n\n”);
printf(“complete name after concatenation\n”);
printf(“%s”, name);
getche();
}
```

Output:

```
First name: ANANDA
Second name: MURUGAN
Last name: SELVARAJ

Complete name after concatenation

ANANDAMURUGANSELVARAJ
```

84. Write a C program to append second string with specified (n) number of characters at the end of the first string.

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
Char text1[30], text2[10],n;
Puts(“Enter text1:”);
gets(text1)
puts(“Enter text2:”);
gets(text2);
printf(“Enter number of characters  to add:”);
gets(n);
strcat(text1,””);
strncat(text1, text2,n);
clrscr();
printf(“%s\n”, text1);
getch();
}
```

Output:

```
Enter  text1:MAY  I
Enter text2: COME IN?
Enter number of characters to add:4
MAY I COME
```

85. Write a C program to display reverse of the given string.

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
Char text[15];
Puts(“Enter string”);
gets(text);
puts(“Reverse string”);
puts(strrev(text));
getch();
}
```

Output:

```
Enter  string

ANANDAMURUGAN

Reverse string

NAGURUMADNANA
```

86. Write a C program to replace a given string with a given symbol.

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char  text[15];
char   symbol;
clrscr();
Puts(“Enter string”);
gets(string);
puts(“Enter symbol for replacement:”);
scanf(“%c”, &symbol);
printf(“Before strset(): %s\n”, string);
strset(string, symbol);
printf(“After strset(): %s\n”,r, string);
getch();
}
```

Output:

```
Enter  string: SAM
Enter symbol for replacement: A
Before strset(): SAM
After strset(): AAA
```

87. Write a C program to replace a given string with a given symbol for the given number of arguments.

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char string[15]; char symbol;
int n;
clrscr();
Puts(“Enter string”);
gets(string);
puts(“Enter symbol for replacement:”);
scanf(“%c”, &symbol);
puts(“how many string character to be replaced”);
scanf(“%d”, &n);
printf(“Before strset(): %s\n”, string);
strset(string, symbol);
printf(“After strset(): %s\n”, string);
getch();
}
```

Output:

```
Enter  string: ANANDAMURUGAN
Enter symbol for replacement: +
How many string characters to be replaced 4
Before strset(): ANANDAMURUGAN
After strset(): ++++DAMURUGAN
```

88. Write a C program to enter two strings. Indicate after what character the lengths of the two strings have no match.

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char stra[10], strb[10];
int length;
clrscr();
Printf("First string:");
gets(stra);
printf("second string:");
gets(strb);
length=strspn(stra, strb);
printf("After %d characters there is no match\n, length");
getch();
}
```

Output:

```
First string: GOOD MORNING
Second string: GOOD BYE
After 5 characters there is no match
```

89. Write a C program to print the given string from the first occurrence of a given character.

```
 #include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char *ptr; char text1[20], text2[2];
clrscr();
printf("Enter string:");
gets(text1);
printf("Enter character:");
gets(text2);
ptr=strpbrk (text1, text2);
puts("string from given character");
printf(ptr);
getch();
}
```

Output:

```
Enter a string: INDIA IS GREAT
Enter character :G
String from given character: GREAT
```

90. Write a C program to add two numbers through variables and their pointers.

```
 #include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c,d,*ap,*bp;
clrscr();
printf("Enter two numbers:");
scanf("%d%d", &a,&b);
ap=&a;
bp=&b;
c=a+b;
d=*ap+*bp;
printf("\n sum of A &B using variable:%d",c);
printf("\n sum of A &B using pointer:%d",d);
getch();
}
```

Output:

```
Enter two numbers: 8 4
Sum of a & B using variable:12
Sum of a & B using pointer:12
```

92. Write a C program to find length of a given string including and executing spaces using pointers.

```
 #include<stdio.h>
#include<conio.h>
void main()
{
Char str[20], *s;
int p=0, q=0;
clrscr();
printf(“Enter String:”);
gets(str); s=str;
while(*s!=‟\0‟)
{
Printf(“%c”, *s);
P++;
S++;
If(*s= =32)
q++;
}
printf(“\n Length of string including spaces: %d”, p);
printf(“\n Length of string excluding spaces: %d”, p-q);
getch();
}
```

Output:

```
Enter string: POINTER ARE EASY
POINTERS ARE EASY

 Length of string including spaces: 17
Length of string excluding spaces: 15
```

93. Write a C program to interchange the elements of a character array using pointer.

```
 #include<stdio.h>
#include<conio.h>
void main()
{
char  *names[]={“anand”,“murugan”,“selvaraj”, “annammal”,”shrikarthick”, “renukadevi”};
char*tmp;
clrscr();
printf(“original: %s%s”, names[3], names[4]);
tmp=names[3];
names[3]=names[4];
names[4]=temp;
printf(“\n New : %s%s”,names[3],names[4]);
getch();
}
```

Output:

```
Original : annammal          shrikarthick
New       :shrikarthick        annammal
```

94. Write a C program to copy structure elements from one object to another object

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

```
struct disk
{
char co[15];
float type;
int price;
};
Struct disk d1={“SONY”,1.44,20};
Struct disk d2,d3;
strcpy(d2.co,d1.co);
d2.type=d1.type;
d2.price=d1.price;
d3=d2;
clrscr();
printf(“\n %s %g %d ”, d1.co.d1.type, d1.price);
printf(“\n %s %g %d ”, d2.co.d2.type, d2.price);
printf(“\n %s %g %d ”, d3.co.d3.type, d3.price);
getch();
}
```

Output:

```
SONY 1.44 20
SONY 1.44 20
SONY 1.44 20
```

95. Write a C program to create the user defined data type hours on int data type and use it in the program

```
 #include<stdio.h>
#include<conio.h>
#define H 60
void main()
{
typedef  int hours;
hours hrs;
clrscr();
printf(“Enter hours:”); scanf(“%d”,
&hrs); printf(“\nMinutes:
%d”,hrs*H); printf(“\n Seconds:
%d”,hrs*H*H); getch();
}
```

Output:

```
Enter hours: 2
Minutes: 120
Seconds: 7200
```

96. Write a C program to store the information of vehicles. Use bit fields to store the status of information

```
 #include<stdio.h>
#include<conio.h>
#define PETROL 1
#define DISEL 2
#define TWO_WH 3
#define FOUR_WH 4
#define OLD 5
#define NEW 6
void main()
{
struct vehicle
{
unsigned type:3;
unsigned fuel:2;
unsigned model:3;
};
struct vehicle v;
v.type=FOUR_WH;
v.fuel=DISEL;
v.model=OLD;
clrscr();
printf("\n Type of vehicle: %d",v.type);
printf("\n Fuel: %d",v.fuel);
printf("\n Model: %d",v.model);
getch();
}
```

Output:

```
Type of Vehicle : 4
Fuel            : 2
Model           : 5
```

97. Write a C program to create enumerated data type for 12 months. Display their values in integer constants

```
 #include<stdio.h>
#include<conio.h>
void main()
{
Enum month{Jan, Feb, mar, Apr, May, June, July, Aug, Sep, Oct, Nov, Dec}
clrscr();
printf("\n Jan=%d", Jan);
printf("\n Feb=%d", Feb);
printf("\n June=%d", June);
printf("\n Dec=%d", Dec);
getch();
}
```

Output:

```
Jan=0
Feb=1
June=5
Dec=11
```

98. Write a C program to open a file in read/write mode. Read and Write new information in the file.

```
 #include<stdio.h>
#include<conio.h>
#include<process.h>
void main()
{
FILE   *fp;
char  c=‟ „;
clrscr();
fp=fopen("data.txt", "r+");
if(fp= = NULL)
{
Printf("cannot open file");
exit(1);
}
Printf("\n Contents read:");
While(!feof(fp))
Printf("%c" getc(fp));
Printf("write data & to stop press „.‟ :");
while(c=‟!‟)
{
C=getche();
fputc(c,fp);
}
getch();
}
```

Output:

```
Contents read: Help me.
Write data & to stop press‟.‟: I am in trouble.
```

99. Write a C program to open a file for read/write operation in binary mode. Read and Write new information in the file.

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

```
#include<conio.h>
#include<process.h>
void main()
{
FILE   *fp;
char  c=‟ „;
clrscr();
fp=fopen(“data.dat”, “wb”);
if(fp= = NULL)
{
Printf(“cannot open file”);
exit(1);
}
Printf(“\n Contents read:”);
While(!feof(fp))
Printf(“%c” getc(fp));
Printf(“write data & to stop press „.‟ :”);
while(c=‟!‟)
{
C=getche();
fputc(c,fp);
} fclose(fp);
fp=fopen(“data.dat”, “rb”);
printf(“\n contents read:”);
while(!feof(fp))
printf(“%c”, getc(fp));
getch();
}
```

Output:

```
Contents read: Help me.
Write data & to stop press‟.‟: I am in trouble.
```

100. Write a C program to display number of arguments and their names.

```
 #include<stdio.h>
#include<conio.h>
main(int argc, char *argv[])
{
Int x;
clrscr();
Printf(“\n Total number of arguments are %d \n ”, argc);
for(x=0; x<argc; x++)
Printf(“%s\t”, argv[x]);
getch();
return 0;
}
```

Output:

```
Total numbers of arguments are 4
C:\TC\C.EXE A B C
```