1) An **Armstrong number** of three digits is an integer such that the sum of the cubes of its digits is equal to the **number** itself. For example, 371 is an **Armstrong number** since 3**3 + 7**3 + 1**3 = 371. Write a program to find all **Armstrong number** in the range of 0 and 999.

2) A palindromic number is a number (in some base) that is the same when written forwards or backwards, i.e., of the form . The first few palindromic numbers are therefore are **0**, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99, **101**, 111, **121**, .

1. Write a program to find factorial of the given number...
2. Write a program to check whether the given number is even or odd.
3. Write a program to swap two numbers using a temporary variable.
4. Write a program to swap two numbers without using a temporary variable.
5. Write a program to swap two numbers using bitwise operators.
6. Write a program to find the greatest of three numbers.
7. Write a program to find the greatest among ten numbers.
8. Write a program to check whether the given number is a prime.
9. Write a program to check whether the given number is a palindrome c number.
10.Write a program to check whether the given string is a palindrome .
11.Write a program to generate the Fibonacci series.
12.Write a program to print"Hello World"without using semicolon anywhere in the code.
13.Write a program to print a semicolon without using a semicolon anywhere in the code.
14.Write a program to compare two strings without using strcmp() function.
15.Write a program to concatenat e two strings without using strcat() function.
16.Write a program to delete a specified line from a text file.
17.Write a program to replace a specified line in a text file.
18.Write a program to find the number of lines in a text file..
19.Write a C program which asks the user for a number between 1 to 9 and shows the number. If the user
inputs a number out of the specified range, the program should show an error and prompt the user for a
valid input.
20.Write a program to display the multiplica tion table of a given number..
21.WAP to check a string is Caliondrom e or not. // Maventic question.
22.WAP to print DONE,witho ut using any loop. // asked to my frnd in any company.
23.WAP to print DONE,witho ut using any loop and any conditonal clause or operators. // asked to me as a cross question of 22th question by the person i asked 22th ques.
24. WAP to find out the longest word in a string.
25.Prog of WORLD MAP. // this code was written by someone,i forgot his name,he won award for this code as short and best c code. JUST FOR FUN //
26.WAP to print the triangle of letters in increasing order of lines..
27.WAP to print'xay'in place of every'a'in a string.// DOC Update on 24-jan-12.
28.Count the Total Number of 7 comming between 1 to 100.
/* I made this code in a way that u can give Upper limit i.e. 100,Lower limit i.e. 1 and the specific

number u wants to count in between i.e. 7 */ // asked by: Vishwa Pratap Rana..
29. Code for duplicate' s removal,by Amit Aru.. // Similar question was asked in Maventic 2nd round to me,,
30. WAP to find out if a given number is a power series of 2 or not,withou t any loop and without using % modulo operator.. // asked by someone on BJS..

TO BE CONTINUED. ..!!!

ANSWERS

1. Write a program to find factorial of the given number.
Recursion: A function is called'recursive 'if a statement within the body of a function calls the same function. It
is also called'circular definition '. Recursion is thus a process of defining something in terms of itself.
Program: To calculate the factorial value using recursion.

```
#include
int fact(int n);
int main(){
int x, i;
printf("En ter a value for x: \n");
scanf("%d" ,&x);
i = fact(x);
printf("\n Factorial of %d is %d", x, i);
return 0;
}int fact(int n){
/* n=0 indicates a terminatin g condition */
if (n
return (1);
}else{
/* function calling itself */
return (n * fact(n - 1));
/*n*fact(n -1) is a recursive expression */
}
}
```

Output:

```
Enter a value for x:
4
Factorial of 4 is 24
```

Explanatio n:

```
fact(n) = n * fact(n-1)
If n=4
fact(4) = 4 * fact(3) there is a call to fact(3)
fact(3) = 3 * fact(2)
fact(2) = 2 * fact(1)
fact(1) = 1 * fact(0)
fact(0) = 1
```

fact(1) = 1 * 1 = 1
fact(2) = 2 * 1 = 2
fact(3) = 3 * 2 = 6
Thus fact(4) = 4 * 6 = 24
Terminatin g condition(n
infinite loop.

2. Write a program to check whether the given number is even or odd.
Program:

```
#include
int main(){
int a;
printf("En ter a: \n");
scanf("%d" ,&a);
/* logic */
if (a % 2 == 0){
printf("Th e given number is EVEN\n");
}
else{
printf("Th e given number is ODD\n");
}
return 0;
}
```

Output:

```
Enter a: 2
The given number is EVEN
```

Explanatio n with examples:
Example 1: If entered number is an even number
Let value of'a'entered is 4
if(a%2==0) then a is an even number, else odd.
i.e. if(4%2==0) then 4 is an even number, else odd.
To check whether 4 is even or odd, we need to calculate (4%2).
/* % (modulus) implies remainder value. */
/* Therefore if the remainder obtained when 4 is divided by 2 is 0, then 4 is even. */
4%2==0 is true
Thus 4 is an even number.
Example 2: If entered number is an odd number.
Let value of'a'entered is 7
if(a%2==0) then a is an even number, else odd.
i.e. if(7%2==0) then 4 is an even number, else odd.
To check whether 7 is even or odd, we need to calculate (7%2).
7%2==0 is false /* 7%2==1 condition fails and else part is executed */
Thus 7 is an odd number.

3. Write a program to swap two numbers using a temporary variable.
Swapping interchang es the values of two given variables.

Logic:

```
step1: temp=x;
step2: x=y;
step3: y=temp;
```

Example:

if x=5 and y=8, consider a temporary variable temp.

```
step1: temp=x=5;
step2: x=y=8;
step3: y=temp=5;
```

Thus the values of the variables x and y are interchang ed.

Program:

```
#include
int main(){
int a, b, temp;
printf("En ter the value of a and b: \n");
scanf("%d %d",&a,&b);
printf("Be fore swapping a=%d, b=%d \n", a, b);
/*Swapping logic */
temp = a;
a = b;
b = temp;
printf("Af ter swapping a=%d, b=%d", a, b);
return 0;
}
```

Output:

```
Enter the values of a and b: 2 3
Before swapping a=2, b=3
After swapping a=3, b=2
```

4. Write a program to swap two numbers without using a temporary variable.

Swapping interchang es the values of two given variables.

Logic:

```
step1: x=x+y;
step2: y=x-y;
step3: x=x-y;
```

Example:

if x=7 and y=4

```
step1: x=7+4=11;
step2: y=11-4=7;
step3: x=11-7=4;
```

Thus the values of the variables x and y are interchang ed.

Program:

```
#include
int main(){
int a, b;
printf("Enter values of a and b: \n");
scanf("%d %d",&a,&b);
printf("Before swapping a=%d, b=%d\n", a,b);
/*Swapping logic */
a = a + b;
b = a - b;
a = a - b;
printf("After swapping a=%d b=%d\n", a, b);
return 0;
}
```

Output:

```
Enter values of a and b: 2 3
Before swapping a=2, b=3
The values after swapping are a=3 b=2
```

5. Write a program to swap two numbers using bitwise operators.

Program:

```
#include
int main(){
int i = 65;
int k = 120;
printf("\n value of i=%d k=%d before swapping", i, k);
i = i ^ k;
k = i ^ k;
i = i ^ k;
printf("\n value of i=%d k=%d after swapping", i, k);
return 0;
}
```

Explanation:

```
i = 65; binary equivalent of 65 is 0100 0001
k = 120; binary equivalent of 120 is 0111 1000
i = i^k;
i...0100 0001
k...0111 1000
---------
val of i = 0011 1001
---------
k = i^k
i...0011 1001
k...0111 1000
---------
val of k = 0100 0001 binary equivalent of this is 65
---------( that is the initial value of i)
i = i^k
i...0011 1001
```

```
k...0100 0001
---------
val of i = 0111 1000 binary equivalent of this is 120
--------- (that is the initial value of k)
```

6. Write a program to find the greatest of three numbers.

Program:

```
#include
int main(){
int a, b, c;
printf("En ter a,b,c: \n");
scanf("%d %d %d",&a,&b,&c);
if (a>b&&a>c){
printf("a is Greater than b and c");
}
else if (b>a&&b>c){
printf("b is Greater than a and c");
}
else if (c>a&&c>b){
printf("c is Greater than a and b");
}
else{
printf("al l are equal or any two values are equal");
}
return 0;
}
```

Output:

```
Enter a,b,c: 3 5 8
c is Greater than a and b
```

Explanatio n with examples:

```
Consider three numbers a=5,b=4,c= 8
if(a>b&&a>c) then a is greater than b and c
now check this condition for the three numbers 5,4,8 i.e.
if(5>4&&5>8) /* 5>4 is true but 5>8 fails */
so the control shifts to else if condition
else if(b>a&&b>c) then b is greater than a and c
now checking this condition for 5,4,8 i.e.
else if(4>5&&4>8) / * both the conditions fail */
now the control shifts to the next else if condition
else if(c>a&&c>b) then c is greater than a and b
now checking this condition for 5,4,8 i.e.
else if(8>5&&8>4) / * both conditions are satisfied */
Thus c is greater than a and b.
```

7. Write a program to find the greatest among ten numbers.

Program:

```
#include
int main(){
int a[10];
int i;
int greatest;
printf("En ter ten values:");
//Store 10 numbers in an array
for (i = 0; i<10; i++){
scanf("%d" ,&a[i]);
}
//Assume that a[0] is greatest
greatest = a[0];
for (i = 0; i<10; i++){
if (a[i]>greatest){
greatest = a[i];
}
}
printf("\n Greatest of ten numbers is %d", greatest);
return 0;
}
```

Output:

```
Enter ten values: 2 53 65 3 88 8 14 5 77 64 Greatest of ten numbers is 88
```

Explanatio n with example:

Entered values are 2, 53, 65, 3, 88, 8, 14, 5, 77, 64

They are stored in an array of size 10. let a[] be an array holding these values.

/* how the greatest among ten numbers is found */

Let us consider a variable'greatest' . At the beginning of the loop, variable'greatest' is assinged with the value of first element in the array greatest=a [0]. Here variable'greatest' is assigned 2 as a[0]=2.

Below loop is executed until end of the array'a[]';.

```
for(i=0; i
{
if(a[i]>gr eatest)
{
greatest= a[i];
}
}
```

For each value of'i', value of a[i] is compared with value of variable'greatest' . If any value greater than the value of'greatest' is encountere d, it would be replaced by a[i]. After completion of'for'loop, the value of variable 'greatest' holds the greatest number in the array. In this case 88 is the greatest of all the numbers.

8. Write a program to check whether the given number is a prime.

A prime number is a natural number that has only one and itself as factors. Examples: 2, 3, 13 are prime

numbers.

Program:

```
#include
main(){
int n, i, c = 0;
printf("En ter any number n: \n");
scanf("%d" ,&n);
/*logic*/
for (i = 1; i
if (n % i == 0){
c++;
}
}
if (c == 2){
printf("n is a Prime number");
}
else{
printf("n is not a Prime number");
}
return 0;
}
```

Output:

```
Enter any number n: 7
n is Prime
```

Explanatio n with examples:

consider a number n=5

for(i=0;i

i.e. for(i=0;i

1st iteration: i=1;i

here i is incremente d i.e. i value for next iteration is 2

now if(n%i==0) then c is incremente d

i.e.if(5%1 ==0)then c is incremente d, here 5%1=0 thus c is incremente d.

now c=1;

2nd iteration: i=2;i

here i is incremente d i.e. i value for next iteration is 3

now if(n%i==0) then c is incremente d

i.e.if(5%2 ==0) then c is incremente d, but 5%2!=0 and so c is not incremente d, c remains 1

c=1;

3rd iteration: i=3;i

here i is incremente d i.e. i value for next iteration is 4

now if(n%i==0) then c is incremente d

i.e.if(5%3 ==0) then c ic incremente d, but 5%3!=0 and so c is not incremente d, c remains 1

c=1;

4th iteration: i=4;i

here i is incremente d i.e. i value for next iteration is 5

now if(n%i==0) then c is incremente d

i.e. if(5%4==0) then c is incremente d, but 5%4!=0 and so c is not incremente d, c remains 1

c=1;

5th iteration: i=5;i

here i is incremente d i.e. i value for next iteration is 6

now if(n%i==0) then c is incremente d

i.e. if(5%5==0) then c is incremente d, 5%5=0 and so c is incremente d.

i.e. c=2

6th iteration: i=6;i

here i value is 6 and 6

now if(c==2) then n is a prime number

we have c=2 from the 5th iteration and thus n=5 is a Prime number.

9. Write a program to check whether the given number is a palindromi c number.

If a number, which when read in both forward and backward way is same, then such a number is called a

palindrome number.

Program:

```
#include
int main(){
int n, n1, rev = 0, rem;
printf("En ter any number: \n");
scanf("%d" ,&n);
n1 = n;
/* logic */
while (n>0){
rem = n % 10;
rev = rev * 10 + rem;
n = n / 10;
}
if (n1 == rev){
printf("Gi ven number is a palindromi c number");
}
else{
printf("Gi ven number is not a palindromi c number");
}
return 0;
}
```

Output:

```
Enter any number: 121
Given number is a palindrome
```

Explanatio n with an example:

Consider a number n=121, reverse=0, remainder;

number=121

now the while loop is executed /* the condition (n>0) is satisfied */

/* calculate remainder */

remainder of 121 divided by 10=(121%10)=1;

now reverse=(r everse*10) +remainder

=(0*10)+1 / * we have initialized reverse=0 */

=1

number=num ber/10

=121/10

=12

now the number is 12, greater than 0. The above process is repeated for number=12.

remainder= 12%10=2;

reverse=(1 *10)+2=12;

number=12/ 10=1;

now the number is 1, greater than 0. The above process is repeated for number=1.

remainder= 1%10=1;

reverse=(1 2*10)+1=12 1;

number=1/ 10 / * the condition n>0 is not satisfied,co ntrol leaves the while loop */

Program stops here. The given number=121 equals the reverse of the number. Thus the given number is a palindrome number.

10.Write a program to check whether the given string is a palindrome .

Palindrome is a string, which when read in both forward and backward way is same.

Example: radar, madam, pop, lol, rubber, etc.,

Program:

```
#include
#include
int main(){
char string1[20 ];
int i, length;
int flag = 0;
printf("En ter a string: \n");
scanf("%s" , string1);
length = strlen(str ing1);
for(i=0;i<length ;i++){
if(string1 [i] != string1[le ngth-i-1]) {
flag = 1;
break;
}
}
if (flag){
printf("%s is not a palindrome \n", string1);
}
else{
```

```
printf("%s is a palindrome \n", string1);
}
return 0;
}
```

Output:

```
Enter a string: radar
"radar"is a palindrome
```

Explanatio n with example:

To check if a string is a palindrome or not, a string needs to be compared with the reverse of itself.

Consider a palindrome string:"radar",

```
---------- ---------- -------
index: 0 1 2 3 4
value: r a d a r
---------- ---------- -------
```

To compare it with the reverse of itself, the following logic is used:

0th character in the char array, string1 is same as 4th character in the same string.

1st character is same as 3rd character.

2nd character is same as 2nd character.

. . . .

ith character is same as'length-i- 1'th character.

If any one of the above condition fails, flag is set to true(1), which implies that the string is not a palindrome .

By default, the value of flag is false(0). Hence, if all the conditions are satisfied, the string is a palindrome .

11.Write a program to generate the Fibonacci series.

Fibonacci series: Any number in the series is obtained by adding the previous two numbers of the series.

Let f(n) be n'th term.

f(0)=0;

f(1)=1;

f(n)=f(n-1)+f(n-2); (for n>=2)

Series is as follows

```
011
(1+0)
2 (1+1)
3 (1+2)
5 (2+3)
8 (3+5)
13 (5+8)
21 (8+13)
34 (13+21)
...and so on
```

Program: to generate Fibonacci Series(10 terms)

```
#include
```

```
int main(){
//array fib stores numbers of fibonacci series
int i, fib[25];
// initialized first element to 0
fib[0] = 0;
// initialized second element to 1
fib[1] = 1;
//loop to generate ten elements
for (i = 2; i<10; i++){
//i'th element of series is equal to the sum of i-1'th element and i-2'th element.
fib[i] = fib[i - 1] + fib[i - 2];
}
printf("Th e fibonacci series is as follows \n");
//print all numbers in the series
for (i = 0; i<10; i++){
printf("%d \n", fib[i]);
}
return 0;
}
```

Output:

```
The fibonacci series is as follows
01123581
3
21
34
```

Explanatio n:

The first two elements are initialize d to 0, 1 respective ly. Other elements in the series are generated by looping
and adding previous two numbes. These numbers are stored in an array and ten elements of the series are
printed as output.

12.Write a program to print"Hello World"without using semicolon anywhere in the code.

Generally when we use printf("") statement, we have to use a semicolon at the end. If printf is used inside an if
Condition, semicolon can be avoided.

Program: Program to print something without using semicolon (;)

```
#include
int main(){
//printf returns the length of string being printed
if (printf("H ello World\n")) //prints Hello World and returns 11
{
//do nothing
}
return 0;
}
```

Output:

```
Hello World
```

Explanatio n:

The if statement checks for condition whether the return value of printf("He llo World") is greater than 0. printf
function returns the length of the string printed. Hence the statement if (printf("H ello World")) prints the string
"Hello World".

13.Write a program to print a semicolon without using a semicolon anywhere in the code.
Generally when use printf("") statement we have to use semicolon at the end.
If we want to print a semicolon, we use the statement: printf(";");
In above statement, we are using two semicolons . The task of printing a semicolon without using semicolon anywhere in the code can be accomplish ed by using the ascii value of';'which is equal to 59.
Program: Program to print a semicolon without using semicolon in the code.

```
#include
int main(void) {
//prints the character with ascii value 59, i.e., semicolon
if (printf("% c\n", 59)){
//prints semicolon
}
return 0;
}
```

Output:

```
;
```

Explanatio n:

If statement checks whether return value of printf function is greater than zero or not. The return value of function
call printf("%c ",59) is 1. As printf returns the length of the string printed. printf("%c ",59) prints ascii value that
correspond s to 59, that is semicolon(.

14.Write a program to compare two strings without using strcmp() function.
strcmp() function compares two strings lexicograp hically. strcmp is declared in stdio.h
Case 1: when the strings are equal, it returns zero.
Case 2: when the strings are unequal, it returns the difference between ascii values of the characters that differ.
a) When string1 is greater than string2, it returns positive value.
b) When string1 is lesser than string2, it returns negative value.
Syntax:

```
int strcmp (const char *s1, const char *s2);
```

Program: to compare two strings.

```
#include
#include
```

```
int cmpstr(cha r s1[10], char s2[10]);
int main(){
char arr1[10] ="Nodalo";
char arr2[10] ="nodalo";
printf("%d", cmpstr(arr 1, arr2));
// cmpstr() is equivalent of strcmp()
return 0;
}/
/s1, s2 are strings to be compared
int cmpstr(cha r s1[10], char s2[10]){
//strlen function returns the length of argument string passed
int i = strlen(s1) ;
int k = strlen(s2) ;
int bigger;
if (i<k){
bigger = k;
}
else if (i>k){
bigger = i;
}
else{
bigger = i;
}
//loops'bigger'times
for (i = 0; i<bigger; i++){
// if ascii values of characters s1[i], s2[i] are equal do nothing
if (s1[i] == s2[i]){
}
//else return the ascii difference
else{
return (s1[i] - s2[i]);
}
}
//return 0 when both strings are same
//This statement is executed only when both strings are equal
return (0);
}
```

Output:

```
-32
```

Explanatio n:

cmpstr() is a function that illustrate s C standard function strcmp(). Strings to be compared are sent as arguments to cmpstr().

Each character in string1 is compared to its correspond ing character in string2. Once the loop encounters a differing character in the strings, it would return the ascii difference of the differing characters and exit.

15.Write a program to concatenat e two strings without using strcat() function.

strcat(str ing1,strin g2) is a C standard function declared in the header file string.h

The strcat() function concatenat es string2, string1 and returns string1.

Program: Program to concatenat e two strings

```
#include
#include
char *strct(cha r *c1, char *c2);
char *strct(cha r *c1, char *c2){
//strlen function returns length of argument string
int i = strlen(c1) ;
int k = 0;
// loops until null is encountered and appends string c2 to c1
while (c2[k] !='\0'){
c1[i + k] = c2[k];
k++;
}
return c1;
}
int main(){
char string1[15 ] ="first";
char string2[15 ] ="second";
char *finalstr;
printf("Be fore concatenat ion:"
"\n string1 = %s \n string2 = %s", string1, string2);
// addresses of string1, string2 are passed to strct()
finalstr = strcat(str ing1, string2);
printf("\n After concatenat ion:");
//prints the contents of string whose address is in finalstr
printf("\n finalstr = %s", finalstr);
//prints the contents of string1
printf("\n string1 = %s", string1);
//prints the contents of string2
printf("\n string2 = %s", string2);
return 0;
}
```

Output:

```
Before concatenat ion:
string1 = first
string2 = second
After concatenat ion:
finalstr = firstsecon d
```

```
string1 = firstsecon d
string2 = second
```

Explanatio n:

string2 is appended at the end of string1 and contents of string2 are unchanged.
In strct() function, using a for loop, all the characters of string'c2'are copied at the end of c1. return (c1) is
equivalent to return&c1[0] and it returns the base address of'c1'.'finalstr' stores that address returned by the
function strct().

16.Write a program to delete a specified line from a text file.

In this program, user is asked for a filename he needs to change. User is also asked for the line number that is
to be deleted. The filename is stored in'filename' . The file is opened and all the data is transferre d to another file
except that one line the user specifies to delete.

Program: Program to delete a specific line.

```
#include
int main(){
FILE *fp1, *fp2;
// consider 40 character string to store filename
char filename[4 0];
char c;
int del_line, temp = 1;
//asks user for file name
printf("En ter file name:");
// receives file name from user and stores in'filename'
scanf("%s" , filename);
//open file in read mode
fp1 = fopen(file name,"r");
c = getc(fp1);
//until the last character of file is obtained
while (c != EOF)
{
printf("%c ", c);
//print current character and read next character
c = getc(fp1);
}
//rewind
rewind(fp1 );
printf("\n Enter line number of the line to be deleted:") ;
//accept number from user.
scanf("%d" ,&del_line) ;
//open new file in write mode
fp2 = fopen("cop y.c","w");
c = getc(fp1);
while (c != EOF){
c = getc(fp1);
if (c =='\n')
temp++;
//except the line to be deleted
if (temp != del_line)
{
//copy all lines in file copy.c
putc(c, fp2);
}
}
//close both the files.
fclose(fp1 );
fclose(fp2 );
//remove original file
remove(fil ename);
//rename the file copy.c to original name
rename("co py.c", filename);
printf("\n The contents of file after being modified are as follows:\n ");
fp1 = fopen(file name,"r");
c = getc(fp1);
while (c != EOF){
printf("%c ", c);
c = getc(fp1);
}
fclose(fp1 );
return 0;
}
```

Output:

```
Enter file name:abc.t xt
hi.
Hello
how are you?
I am fine
hope the same
Enter line number of the line to be deleted:4
The contents of file after being modified are as follows:
hi.
hello
how are you?
hope the same
```

Explanatio n:

In this program, user is asked for a filename that needs to be modified. Entered file name is stored in a char array'filename' . This file is opened in read mode using file pointer'fp1'. Character'c'is used to read characters from the file and print them to the output. User is asked for the line number in the file to be deleted. The file pointer is rewinded back and all the lines of the file except for the line to be deleted are copied into another file "copy.c". Now"copy.c"is renamed to the original filename. The original file is opened in read mode and the modified contents of the file are displayed on the screen.

17.Write a program to replace a specified line in a text file.

Program: Program to replace a specified line in a text file.

```
#include
int main(void) {
FILE *fp1, *fp2;
// 'filename'i s a 40 character string to store filename
char filename[4 0];
char c;
int del_line, temp = 1;
//asks user for file name
printf("En ter file name:");
// receives file name from user and stores in'filename'
scanf("%s" , filename);
fp1 = fopen(file name,"r");
//open file in read mode
c = getc(fp1);
//print the contents of file .
while (c != EOF){
printf("%c ", c);
c = getc(fp1);
}
//ask user for line number to be deleted.
printf("\n Enter line number to be deleted and replaced") ;
scanf("%d" ,&del_line) ;
//take fp1 to start point.
rewind(fp1 );
//open copy.c in write mode
fp2 = fopen("cop y.c","w");
c = getc(fp1);
while (c != EOF){
if (c =='\n'){
temp++;
}
// till the line to be deleted comes,copy the content from one file to other
if (temp != del_line){
putc(c, fp2);
}
else //when the line to be deleted comes
{
while ((c = getc(fp1)) !='\n'){
}
//read and skip the line ask for new text
printf("En ter new text");
//flush the input stream
fflush(std in);
putc('\n', fp2);
//put'\n'in new file
while ((c = getchar()) !='\n')
putc(c, fp2);
//take the data from user and place it in new file
fputs("\n" , fp2);
temp++;
}
// continue this till EOF is encountere d
c = getc(fp1);
}
//close both files
fclose(fp1 );
fclose(fp2 );
//remove original file
remove(fil ename);
//rename new file with old name opens the file in read mode
rename("co py.c", filename);
fp1 = fopen(file name,"r");
//reads the character from file
c = getc(fp1);
// until last character of file is encountered
while (c != EOF){
printf("%c ", c);
// all characters are printed
c = getc(fp1);
}
//close the file pointer
fclose(fp1 );
return 0;
}
```

Output:

```
Enter file name:abc.t xt
```

```
hi.
hello
how are you?
hope the same
Enter line number of the line to be deleted and replaced:4
Enter new text: sayonara see you soon
hi.
hello
how are you?
sayonara see you soon
```

Explanatio n:

In this program, the user is asked to type the name of the file. The File by name entered by user is opened in read mode. The line number of the line to be replaced is asked as input. Next the data to be replaced is asked. A new file is opened in write mode named"copy.c". Now the contents of original file are transferre d into new file and the line to be modified is deleted. New data is stored in its place and remaining lines of the original file are also transferre d. The copied file with modified contents is replaced with the original file's name. Both the file pointers are closed and the original file is again opened in read mode and the contents of the original file is printed as output.

18.Write a program to find the number of lines in a text file.

Number of lines in a file can be determined by counting the number of new line characters present.

Program: Program to count number of lines in a file.

```
#include
int main()
/* Ask for a filename and count number of lines in the file*/
{
//a pointer to a FILE structure
FILE *fp;
int no_lines = 0;
// consider 40 character string to store filename
char filename[4 0], sample_chr ;
//asks user for file name
printf("En ter file name:");
// receives file name from user and stores in a string named'filename'
scanf("%s" , filename);
//open file in read mode
fp = fopen(file name,"r");
//get character from file and store in sample_chr
sample_chr = getc(fp);
while (sample_ch r != EOF){
// Count whenever sample_chr is'\n'(new line) is encountere d
if (sample_ch r =='\n')
{
// increment variable'no_lines' by 1
no_lines=n o_lines+1;
}
//take next character from file.
sample_chr = getc(fp);
}
fclose(fp) ; //close file.
printf("Th ere are %d lines in %s \n", no_lines, filename);
return 0;
}
```

Output:

```
Enter file name:abc.t xt
There are 4 lines in abc.txt
```

Explanatio n:

In this program, name of the file to be read is taken as input. A file by the given name is opened in read-mode using a File pointer'fp'. Characters from the file are read into a char variable'sample_ch r'with the help of getc function. If a new line character('\n') is encountere d, the integer variable'no_lines' is incremente d. If the character read into'sample_ch ar'is not a new line character, next character is read from the file. This process is continued until the last character of the file(EOF) is encountere d. The file pointer is then closed and the total number of lines is shown as output.

19.Write a C program which asks the user for a number between 1 to 9 and shows the number. If the user inputs a number out of the specified range, the program should show an error and prompt the user for a valid input.

Program: Program for accepting a number in a given range.

```
#include
int getnumber( );
int main(){
int input = 0;
//call a function to input number from key board
input = getnumber( );
//when input is not in the range of 1 to 9,print error message
while (!((input = 1))){
printf("[E RROR] The number you entered is out of range");
//input another number
input = getnumber( );
}
//this function is repeated until a valid input is given by user.
printf("\n The number you entered is %d", input);
return 0;
}/
/this function returns the number given by user
int getnumber( ){
int number;
//asks user for a input in given range
printf("\n Enter a number between 1 to 9 \n");
scanf("%d" ,&number);
return (number);
}
```

Output:

```
Enter a number between 1 to 9
45
[ERROR] The number you entered is out of range
Enter a number between 1 to 9
4
The number you entered is 4
```

Explanatio n:

getfunctio n() function accepts input from user.'while'loop checks whether the number falls within range or not

and accordingl y either prints the number(If the number falls in desired range) or shows error message(nu mber is

out of range).

20.Write a program to display the multiplica tion table of a given number.

Program: Multiplica tion table of a given number

```
#include
int main(){
int num, i = 1;
printf("\n Enter any Number:");
scanf("%d" ,&num);
printf("Mu ltiplicati on table of %d: \n", num);
while (i
printf("\n %d x %d = %d", num, i, num * i);
i++;
}
return 0;
}
```

Output:

```
Enter any Number:5
5 x 1 = 5
```

```
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

Explanatio n:

We need to multiply the given number (i.e. the number for which we want the multiplica tion table) with value of'i'which increments from 1 to 10.

21. .WAP to check a string is Caliondrom e or not. // Maventic question.

```
#include
#include
void main()
{
int i,j=0; char a[100];
clrscr();
printf("\n Enter the string to check for caliondrom e:\n");
gets(a);
if(strlen( a)%6)
{
printf("\n %s: is Not a caliondrom e..",a);
getch();
exit(0);
}
for (i=0;a[i]! ='\0'
{
if((a[i]== a[i+5])&&( a[i+1]==a[ i+4])&&(a[ i+2]==a[i+ 3]))
i=i+6;
else
{
j=1;
break;
}
}
if(j)
printf("\n %s: is Not a caliondrom e..",a);
else
printf("\n %s: is a caliondrom e..",a);
getch();
}
```

22.WAP to print DONE,witho ut using any loop. // asked to my frnd in any company.

```
#include
void main()
{
static int i=0;
printf("\n %d. DONE",i);
if(i++
main();
getch();
exit(0); / * I used exit(0) to terminate the program after 100 DONE,,i dunno why it was not
terminating without using it,may be just at my system,try without it at ur sustem,it sud work */
}
```

23.WAP to print DONE,witho ut using any loop and any conditonal clause or operators.

```
/* This code is just in purpose to solve the above question,, but its not a good code in programmin
g,as its terminatin g at divide error,,if anyone have a better code,let me know */

main()
{
static int i=100;
printf("%d . DONE\n",10 1-i);
main(1/ --i);
}

/* use"ctrl+f9",then"alt+f5"to see the result */
```

24. WAP to find out the longest word in a string.

```
#include
#include
#include
void main()
{
int i,max=0,co unt=0,j;
char str[100]; / * ={"INDIA IS DEMOCRATIC COUNTRY"}; u can use a string inside,in place of user
input */

printf("\n Enter the string\n:" );
gets(str);

for(i=0;i
{
if(!(str[i ]==32))
{
count++;
}
else
{
```

```
if(max
{
j=i-count;
max=count;
}
count=0;
}
}
for(i=j;i
printf("%c ",str[i]);
getch();
}
```

25.Prog of WORLD MAP.

```
#include main(l ,a,n,d)cha r**a;{for(d=atoi (a[1])/ 10*80- atoi(a[2]) / 5-596;n="@N KA\CLCCGZA
AQBEAADAFa ISADJABBA^ \SNLGAQABD AXIMBAACTB ATAHDBAN\Z cEMMCCCCAA
hEIJFAEAAA BAfHJE\TBd FLDAANEfDN BPHdBcBBBE A_AL\ H E L L O, W O R L D!"[l++-3];)f
or(;n-->64 putchar(!d +++33^ l&1);print f("\n\n\n\ n\t\tFound By:\n\t\t\ t Amit Aru");getc h();}
```

26.WAP to print the triangle of letters in increasing order of lines.

```
#include
#include
void main()
{
int i,j,k;
char ch;
printf("\n Enter the number of lines wants to make the triangle \n:");
scanf("%d" ,&i);
for(j=1;j
{
ch=65;
for(k=1;k
{
printf("%c ",ch++);
}
printf("\n ");
}
getch();
}
```

27.WAP to print'xay'in place of every'a'in a string.

```
#include
#include
void main()
{
int i=0;
char str[100],x ='x',y='y' ;
printf("En ter the string\n:");
gets(str);
while(str[ i]!='\0')
{
if(str[i]= ='a')
{
printf("%c ",x);
printf("%c ",str[i++] );
printf("%c ",y);
}
else
{
printf("%c ",str[i++] );
}
}
getch();
}
```

28.Count the Total Number of 7 comming between 1 to 100.

/* I made this code in a way that u can give Upper limit i.e. 100,Lower limit i.e. 1 and the specific number u wants to count in between i.e. 7 */

```
#include
#include
void main()
{
int i,j,U=100, L=1,count= 0,r=1,n;
clrscr();
printf("\n Enter the number u wants to count\n:");
scanf("%d" ,&n);
printf("\n Enter the lower limit\n:");
scanf("%d" ,&L);
printf("\n Enter the upper limit\n:");
scanf("%d" ,&U);

for (i=L;i
{
j=i;
while(j)
{
r=j%10;
if (r==n)
{
count++;
}
j=j/10;
}
}
if(n==0&&L ==0)
count++;
printf("\n Total Number of %d between %d and %d = %d",n,L,U, count);
getch();
}
```

29. Code for duplicate' s removal,by Amit Aru.

```
#include
#include
void main()
{
int i,j,k=0,co unt[300]={ 0};
char ch,str[100 0],str1[10 00];
clrscr();
printf("\n Enter the string to remove duplicasy\ n:");
gets(str);
for (i=0;str[i ]!='\0';i+ +)
{
ch=str[i];
count["]=0; / * U can use other delimiter inplace of space"here,just put that char inside",for ex:
count['A']=0 ; if u dnt want any delimiter, just remove this line.*/

if(count[c h])
continue;
else
{
str1[k++]= ch;
count[ch]= 1;
}
}
puts(str1) ;
getch();
}
```

30. WAP to find out if a given number is a power series of 2 or not,withou t any loop and without using % modulo operator.

```
#include
#include
int pow2(float );
void main()
{
int i,flag;
clrscr();
```

```
printf("En ter the number\n") ;
scanf("%d" ,&i);
flag=pow2( i);
if(flag)
printf("\n %d is power series of 2",i);
else
printf("\n %d is not a power series of 2",i);
getch();
}
int pow2(float j)
{
static float x;
x=j/2;
if(x==2)
return 1;
if(x
return 0;
x=pow2(x);
}
```

1. Write a program to find factorial of the given number.
2. Write a program to check whether the given number is even or odd.
3. Write a program to swap two numbers using a temporary variable.
4. Write a program to swap two numbers without using a temporary variable.
5. Write a program to swap two numbers using bitwise operators.
6. Write a program to find the greatest of three numbers.
7. Write a program to find the greatest among ten numbers.
8. Write a program to check whether the given number is a prime.
9. Write a program to check whether the given number is a palindromic number.
10.Write a program to check whether the given string is a palindrome.
11.Write a program to generate the Fibonacci series.
12.Write a program to print "Hello World" without using semicolon anywhere in the code.

13.Write a program to print a semicolon without using a semicolon anywhere in the code.
14.Write a program to compare two strings without using strcmp() function.
15.Write a program to concatenate two strings without using strcat() function.
16.Write a program to delete a specified line from a text file.
17.Write a program to replace a specified line in a text file.
18.Write a program to find the number of lines in a text file.
19.Write a C program which asks the user for a number between 1 to 9 and shows the number. If the user inputs a number out of the specified range, the program should show an error and prompt the user for a valid input.
20.Write a program to display the multiplication table of a given number.

ANSWERS

1. Write a program to find factorial of the given number.

Recursion: A function is called 'recursive' if a statement within the body of a function calls the same function. It
is also called 'circular definition'. Recursion is thus a process of defining something in terms of itself.

Program: To calculate the factorial value using recursion.

```
#include <stdio.h>
int fact(int n);
int main() {
int x, i;
printf("Enter a value for x: \n");
scanf("%d", &x);
i = fact(x);
printf("\nFactorial of %d is %d", x, i);
return 0;
} int fact(int n) {
/* n=0 indicates a terminating condition */
if (n <= 0) {
return (1);
} else {
/* function calling itself */
return (n * fact(n - 1));
/*n*fact(n-1) is a recursive expression */
}
}
```

Output:

```
Enter a value for x:
4
Factorial of 4 is 24
```

Explanation:

fact(n) = n * fact(n-1)

If n=4

fact(4) = 4 * fact(3) there is a call to fact(3)

fact(3) = 3 * fact(2)

fact(2) = 2 * fact(1)

fact(1) = 1 * fact(0)

fact(0) = 1

fact(1) = 1 * 1 = 1

fact(2) = 2 * 1 = 2

fact(3) = 3 * 2 = 6

Thus fact(4) = 4 * 6 = 24

Terminating condition(n <= 0 here;) is a must for a recursive program. Otherwise the program enters into an
infinite loop.

Find Us on FACEBOOK …If u like this document Please Like Our PAGE

2. Write a program to check whether the given number is even or odd.

Program:

```
#include <stdio.h>
int main() {
int a;
printf("Enter a: \n");
scanf("%d", &a);
/* logic */
if (a % 2 == 0) {
```

```
printf("The given number is EVEN\n");
}
else {
printf("The given number is ODD\n");
}
return 0;
}
```

Output:

```
Enter a: 2
The given number is EVEN
```

Explanation with examples:

Example 1: If entered number is an even number

Let value of 'a' entered is 4

if(a%2==0) then a is an even number, else odd.

i.e. if(4%2==0) then 4 is an even number, else odd.

To check whether 4 is even or odd, we need to calculate (4%2).

/* % (modulus) implies remainder value. */

/* Therefore if the remainder obtained when 4 is divided by 2 is 0, then 4 is even. */

4%2==0 is true

Thus 4 is an even number.

Example 2: If entered number is an odd number.

Let value of 'a' entered is 7

if(a%2==0) then a is an even number, else odd.

i.e. if(7%2==0) then 4 is an even number, else odd.

To check whether 7 is even or odd, we need to calculate (7%2).

7%2==0 is false /* 7%2==1 condition fails and else part is executed */

Thus 7 is an odd number.

Find Us on FACEBOOK …If u like this document Please Like Our PAGE

3. Write a program to swap two numbers using a temporary variable.

Swapping interchanges the values of two given variables.

Logic:

step1: temp=x;

step2: x=y;

step3: y=temp;

Example:

if x=5 and y=8, consider a temporary variable temp.

step1: temp=x=5;

step2: x=y=8;
step3: y=temp=5;
Thus the values of the variables x and y are interchanged.

Program:

```
#include <stdio.h>
int main() {
int a, b, temp;
printf("Enter the value of a and b: \n");
scanf("%d %d", &a, &b);
printf("Before swapping a=%d, b=%d \n", a, b);
/*Swapping logic */
temp = a;
a = b;
b = temp;
printf("After swapping a=%d, b=%d", a, b);
return 0;
}
```

Output:

```
Enter the values of a and b: 2 3
Before swapping a=2, b=3
After swapping a=3, b=2
```

Find Us on FACEBOOK …If u like this document Please Like Our PAGE

4. Write a program to swap two numbers without using a temporary variable.

Swapping interchanges the values of two given variables.

Logic:

step1: x=x+y;
step2: y=x-y;
step3: x=x-y;

Example:

if x=7 and y=4
step1: x=7+4=11;
step2: y=11-4=7;
step3: x=11-7=4;
Thus the values of the variables x and y are interchanged.

Program:

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

```
int a, b;
printf("Enter values of a and b: \n");
scanf("%d %d", &a, &b);
printf("Before swapping a=%d, b=%d\n", a,b);
/*Swapping logic */
a = a + b;
b = a - b;
a = a - b;
printf("After swapping a=%d b=%d\n", a, b);
return 0;
}
```

Output:

```
Enter values of a and b: 2 3
Before swapping a=2, b=3
The values after swapping are a=3 b=2
```

Find Us on FACEBOOK …If u like this document Please Like Our PAGE

5. Write a program to swap two numbers using bitwise operators.

Program:

```
#include <stdio.h>
int main() {
int i = 65;
int k = 120;
printf("\n value of i=%d k=%d before swapping", i, k);
i = i ^ k;
k = i ^ k;
i = i ^ k;
printf("\n value of i=%d k=%d after swapping", i, k);
return 0;
}
```

Explanation:

i = 65; binary equivalent of 65 is 0100 0001

k = 120; binary equivalent of 120 is 0111 1000

```
i = i^k;
i...0100 0001
k...0111 1000
---------
val of i = 0011 1001
---------
```

```
k = i^k
i...0011 1001
k...0111 1000
---------
val of k = 0100 0001 binary equivalent of this is 65
---------(that is the initial value of i)
i = i^k
i...0011 1001
k...0100 0001
---------
val of i = 0111 1000 binary equivalent of this is 120
--------- (that is the initial value of k)
```

Find Us on FACEBOOK …If u like this document Please Like Our PAGE

6. Write a program to find the greatest of three numbers.

Program:

```
#include <stdio.h>
int main(){
int a, b, c;
printf("Enter a,b,c: \n");
scanf("%d %d %d", &a, &b, &c);
if (a > b && a > c) {
printf("a is Greater than b and c");
}
else if (b > a && b > c) {
printf("b is Greater than a and c");
}
else if (c > a && c > b) {
printf("c is Greater than a and b");
}
else {
printf("all are equal or any two values are equal");
}
return 0;
}
```

Output:

```
Enter a,b,c: 3 5 8
c is Greater than a and b
```

Explanation with examples:

Consider three numbers a=5,b=4,c=8

if(a>b && a>c) then a is greater than b and c

now check this condition for the three numbers 5,4,8 i.e.

if(5>4 && 5>8) /* 5>4 is true but 5>8 fails */

so the control shifts to else if condition

else if(b>a && b>c) then b is greater than a and c

now checking this condition for 5,4,8 i.e.

else if(4>5 && 4>8) /* both the conditions fail */

now the control shifts to the next else if condition

else if(c>a && c>b) then c is greater than a and b

now checking this condition for 5,4,8 i.e.

else if(8>5 && 8>4) /* both conditions are satisfied */

Thus c is greater than a and b.

Find Us on FACEBOOK …If u like this document Please Like Our PAGE

7. Write a program to find the greatest among ten numbers.

Program:

```
#include <stdio.h>
int main() {
int a[10];
int i;
int greatest;
printf("Enter ten values:");
//Store 10 numbers in an array
for (i = 0; i < 10; i++) {
scanf("%d", &a[i]);
}
//Assume that a[0] is greatest
greatest = a[0];
for (i = 0; i < 10; i++) {
if (a[i] > greatest) {
greatest = a[i];
}
}
printf("\nGreatest of ten numbers is %d", greatest);
return 0;
}
```

Output:

```
Enter ten values: 2 53 65 3 88 8 14 5 77 64 Greatest of ten numbers is 88
```

Explanation with example:

Entered values are 2, 53, 65, 3, 88, 8, 14, 5, 77, 64

They are stored in an array of size 10. let a[] be an array holding these values.

/* how the greatest among ten numbers is found */

Let us consider a variable 'greatest'. At the beginning of the loop, variable 'greatest' is assinged with the value of first element in the array greatest=a[0]. Here variable 'greatest' is assigned 2 as a[0]=2.

Below loop is executed until end of the array 'a[]';.

```
for(i=0; i<10; i++)
{
if(a[i]>greatest)
{
greatest= a[i];
}
}
```

For each value of 'i', value of a[i] is compared with value of variable 'greatest'. If any value greater than the value of 'greatest' is encountered, it would be replaced by a[i]. After completion of 'for' loop, the value of variable 'greatest' holds the greatest number in the array. In this case 88 is the greatest of all the numbers.

Find Us on FACEBOOK …If u like this document Please Like Our PAGE

8. Write a program to check whether the given number is a prime.

A prime number is a natural number that has only one and itself as factors. Examples: 2, 3, 13 are prime numbers.

Program:

```
#include <stdio.h>
main() {
int n, i, c = 0;
printf("Enter any number n: \n");
scanf("%d", &n);
/*logic*/
for (i = 1; i <= n; i++) {
if (n % i == 0) {
c++;
}
}
if (c == 2) {
printf("n is a Prime number");
}
else {
printf("n is not a Prime number");
}
return 0;
}
```

Output:

```
Enter any number n: 7
n is Prime
```

Explanation with examples:

consider a number n=5

for(i=0;i<=n;i++) /* for loop is executed until the n value equals i */

i.e. for(i=0;i<=5;i++) /* here the for loop is executed until i is equal to n */

1st iteration: i=1;i<=5;i++

here i is incremented i.e. i value for next iteration is 2

now if(n%i==0) then c is incremented

i.e.if(5%1==0)then c is incremented, here 5%1=0 thus c is incremented.

now c=1;

2nd iteration: i=2;i<=5;i++

here i is incremented i.e. i value for next iteration is 3

now if(n%i==0) then c is incremented

i.e.if(5%2==0) then c is incremented, but 5%2!=0 and so c is not incremented, c remains 1

c=1;

3rd iteration: i=3;i<=5;i++

here i is incremented i.e. i value for next iteration is 4

now if(n%i==0) then c is incremented

i.e.if(5%3==0) then c ic incremented, but 5%3!=0 and so c is not incremented, c remains 1

c=1;

4th iteration: i=4;i<=5;i++

here i is incremented i.e. i value for next iteration is 5

now if(n%i==0) then c is incremented

i.e. if(5%4==0) then c is incremented, but 5%4!=0 and so c is not incremented, c remains 1

c=1;

5th iteration: i=5;i<=5;i++

here i is incremented i.e. i value for next iteration is 6
now if(n%i==0) then c is incremented
i.e. if(5%5==0) then c is incremented, 5%5=0 and so c is incremented.
i.e. c=2
6th iteration: i=6;i<=5;i++
here i value is 6 and 6<=5 is false thus the condition fails and control leaves the for loop.
now if(c==2) then n is a prime number
we have c=2 from the 5th iteration and thus n=5 is a Prime number.

Find Us on FACEBOOK …If u like this document Please Like Our PAGE

9. Write a program to check whether the given number is a palindromic number.

If a number, which when read in both forward and backward way is same, then such a number is called a
palindrome number.

Program:

```
#include <stdio.h>
int main() {
int n, n1, rev = 0, rem;
printf("Enter any number: \n");
scanf("%d", &n);
n1 = n;
/* logic */
while (n > 0){
rem = n % 10;
rev = rev * 10 + rem;
n = n / 10;
}
if (n1 == rev){
printf("Given number is a palindromic number");
}
else{
printf("Given number is not a palindromic number");
}
return 0;
}
```

Output:

```
Enter any number: 121
```

Given number is a palindrome

Explanation with an example:

Consider a number n=121, reverse=0, remainder;

number=121

now the while loop is executed /* the condition (n>0) is satisfied */

/* calculate remainder */

remainder of 121 divided by 10=(121%10)=1;

now reverse=(reverse*10)+remainder

=(0*10)+1 /* we have initialized reverse=0 */

=1

number=number/10

=121/10

=12

now the number is 12, greater than 0. The above process is repeated for number=12.

remainder=12%10=2;

reverse=(1*10)+2=12;

number=12/10=1;

now the number is 1, greater than 0. The above process is repeated for number=1.

remainder=1%10=1;

reverse=(12*10)+1=121;

number=1/10 /* the condition n>0 is not satisfied,control leaves the while loop */

Program stops here. The given number=121 equals the reverse of the number. Thus the given number is a
palindrome number.

Find Us on FACEBOOK …If u like this document Please Like Our PAGE

10.Write a program to check whether the given string is a palindrome.

Palindrome is a string, which when read in both forward and backward way is same.

Example: radar, madam, pop, lol, rubber, etc.,

Program:

```
#include <stdio.h>
#include <string.h>
int main() {
char string1[20];
int i, length;
int flag = 0;
printf("Enter a string: \n");
scanf("%s", string1);
length = strlen(string1);
for(i=0;i < length ;i++){
if(string1[i] != string1[length-i-1]){
flag = 1;
break;
}
}
if (flag) {
printf("%s is not a palindrome\n", string1);
}
else {
printf("%s is a palindrome\n", string1);
}
return 0;
}
```

Output:

```
Enter a string: radar
"radar" is a palindrome
```

Explanation with example:

To check if a string is a palindrome or not, a string needs to be compared with the reverse of itself.

Consider a palindrome string: "radar",

```
---------------------------
index: 0 1 2 3 4
value: r a d a r
---------------------------
```

To compare it with the reverse of itself, the following logic is used:

0th character in the char array, string1 is same as 4th character in the same string.

1st character is same as 3rd character.

2nd character is same as 2nd character.

. . . .

ith character is same as 'length-i-1'th character.

If any one of the above condition fails, flag is set to true(1), which implies that the string is not a palindrome.

By default, the value of flag is false(0). Hence, if all the conditions are satisfied, the string is a palindrome.

11.Write a program to generate the Fibonacci series.

Fibonacci series: Any number in the series is obtained by adding the previous two numbers of the series.

Let f(n) be n'th term.
f(0)=0;
f(1)=1;
f(n)=f(n-1)+f(n-2); (for n>=2)
Series is as follows
011
(1+0)
2 (1+1)
3 (1+2)
5 (2+3)
8 (3+5)
13 (5+8)
21 (8+13)
34 (13+21)
...and so on

Program: to generate Fibonacci Series(10 terms)

```
#include<stdio.h>
int main() {
//array fib stores numbers of fibonacci series
int i, fib[25];
//initialized first element to 0
fib[0] = 0;
//initialized second element to 1
fib[1] = 1;
//loop to generate ten elements
for (i = 2; i < 10; i++) {
//i'th element of series is equal to the sum of i-1'th element and i-2'th element.
fib[i] = fib[i - 1] + fib[i - 2];
}
printf("The fibonacci series is as follows \n");
//print all numbers in the series
for (i = 0; i < 10; i++) {
printf("%d \n", fib[i]);
}
return 0;
}
```

Output:

The fibonacci series is as follows

```
01123581
3
21
34
```

Explanation:

The first two elements are initialized to 0, 1 respectively. Other elements in the series are generated by looping
and adding previous two numbes. These numbers are stored in an array and ten elements of the series are
printed as output.

12.Write a program to print "Hello World" without using semicolon anywhere in the code.

Generally when we use printf("") statement, we have to use a semicolon at the end. If printf is used inside an if
condition, semicolon can be avoided.

Program: Program to print some thing with out using semicolon(;)

```
#include <stdio.h>
int main() {
//printf returns the length of string being printed
if (printf("Hello World\n")) //prints Hello World and returns 11
{
//do nothing
}
return 0;
}
```

Output:

```
Hello World
```

Explanation:

The if statement checks for condition whether the return value of printf("Hello World") is greater than 0. printf
function returns the length of the string printed. Hence the statement if (printf("Hello World")) prints the string
"Hello World".

13.Write a program to print a semicolon without using a semicolon anywhere in the code.

Generally when use printf("") statement we have to use semicolon at the end.
If we want to print a semicolon, we use the statement: printf(";");
In above statement, we are using two semicolons. The task of printing a semicolon without using semicolon
anywhere in the code can be accomplished by using the ascii value of ' ; ' which is equal to 59.

Program: Program to print a semicolon without using semicolon in the code.

```
#include <stdio.h>
int main(void) {
//prints the character with ascii value 59, i.e., semicolon
if (printf("%c\n", 59)) {
//prints semicolon
}
return 0;
}
```

Output:

```
;
```

Explanation:

If statement checks whether return value of printf function is greater than zero or not. The return value of function
call printf("%c",59) is 1. As printf returns the length of the string printed.
printf("%c",59) prints ascii value that
corresponds to 59, that is semicolon(;).

14.Write a program to compare two strings without using strcmp() function.

strcmp() function compares two strings lexicographically. strcmp is declared in stdio.h

Case 1: when the strings are equal, it returns zero.

Case 2: when the strings are unequal, it returns the difference between ascii values of the characters that differ.

a) When string1 is greater than string2, it returns positive value.

b) When string1 is lesser than string2, it returns negative value.

Syntax:

```
int strcmp (const char *s1, const char *s2);
```

Program: to compare two strings.

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

```
#include<string.h>
int cmpstr(char s1[10], char s2[10]);
int main() {
char arr1[10] = "Nodalo";
char arr2[10] = "nodalo";
printf(" %d", cmpstr(arr1, arr2));
//cmpstr() is equivalent of strcmp()
return 0;
}/
/s1, s2 are strings to be compared
int cmpstr(char s1[10], char s2[10]) {
//strlen function returns the length of argument string passed
int i = strlen(s1);
int k = strlen(s2);
int bigger;
if (i < k) {
bigger = k;
}
else if (i > k) {
bigger = i;
}
else {
bigger = i;
}
//loops 'bigger' times
for (i = 0; i < bigger; i++) {
//if ascii values of characters s1[i], s2[i] are equal do nothing
if (s1[i] == s2[i]) {
}
//else return the ascii difference
else {
return (s1[i] - s2[i]);
}
}
//return 0 when both strings are same
//This statement is executed only when both strings are equal
return (0);
}
```

Output:

```
-32
```

Explanation:

cmpstr() is a function that illustrates C standard function strcmp(). Strings to be compared are sent as arguments
to cmpstr().
Each character in string1 is compared to its corresponding character in string2. Once the loop encounters a
differing character in the strings, it would return the ascii difference of the differing characters and exit.

15.Write a program to concatenate two strings without using strcat() function.

strcat(string1,string2) is a C standard function declared in the header file string.h
The strcat() function concatenates string2, string1 and returns string1.

Program: Program to concatenate two strings

```
#include<stdio.h>
#include<string.h>
char *strct(char *c1, char *c2);
char *strct(char *c1, char *c2) {
//strlen function returns length of argument string
int i = strlen(c1);
int k = 0;
//loops until null is encountered and appends string c2 to c1
while (c2[k] != '\0') {
c1[i + k] = c2[k];
k++;
}
return c1;
}
int main() {
char string1[15] = "first";
char string2[15] = "second";
char *finalstr;
printf("Before concatenation:"
" \n string1 = %s \n string2 = %s", string1, string2);
//addresses of string1, string2 are passed to strct()
finalstr = strct(string1, string2);
printf("\nAfter concatenation:");
//prints the contents of string whose address is in finalstr
printf("\n finalstr = %s", finalstr);
//prints the contents of string1
printf("\n string1 = %s", string1);
//prints the contents of string2
printf("\n string2 = %s", string2);
return 0;
}
```

Output:

```
Before concatenation:
string1 = first
string2 = second
After concatenation:
finalstr = firstsecond
string1 = firstsecond
string2 = second
```

Explanation:

string2 is appended at the end of string1 and contents of string2 are unchanged.
In strct() function, using a for loop, all the characters of string 'c2' are copied at the end of c1. return (c1) is equivalent to return &c1[0] and it returns the base address of 'c1'. 'finalstr' stores that address returned by the function strct().

Find Us on FACEBOOK …If u like this document Please Like Our PAGE

16.Write a program to delete a specified line from a text file.

In this program, user is asked for a filename he needs to change. User is also asked for the line number that is to be deleted. The filename is stored in 'filename'. The file is opened and all the data is transferred to another file except that one line the user specifies to delete.

Program: Program to delete a specific line.

```
#include <stdio.h>
int main() {
FILE *fp1, *fp2;
//consider 40 character string to store filename
char filename[40];
char c;
int del_line, temp = 1;
//asks user for file name
printf("Enter file name: ");
//receives file name from user and stores in 'filename'
```

```
scanf("%s", filename);
//open file in read mode
fp1 = fopen(filename, "r");
c = getc(fp1);
//until the last character of file is obtained
while (c != EOF)
{
printf("%c", c);
//print current character and read next character
c = getc(fp1);
}
//rewind
rewind(fp1);
printf(" \n Enter line number of the line to be deleted:");
//accept number from user.
scanf("%d", &del_line);
//open new file in write mode
fp2 = fopen("copy.c", "w");
c = getc(fp1);
while (c != EOF) {
c = getc(fp1);
if (c == '\n')
temp++;
//except the line to be deleted
if (temp != del_line)
{
//copy all lines in file copy.c
putc(c, fp2);
}
}
//close both the files.
fclose(fp1);
fclose(fp2);
//remove original file
remove(filename);
//rename the file copy.c to original name
rename("copy.c", filename);
printf("\n The contents of file after being modified are as follows:\n");
fp1 = fopen(filename, "r");
c = getc(fp1);
```

```
while (c != EOF) {
printf("%c", c);
c = getc(fp1);
}
fclose(fp1);
return 0;
}
```

Output:

```
Enter file name:abc.txt
hi.
Hello
how are you?
I am fine
hope the same
Enter line number of the line to be deleted:4
The contents of file after being modified are as follows:
hi.
hello
how are you?
hope the same
```

Explanation:

In this program, user is asked for a filename that needs to be modified. Entered file name is stored in a char
array 'filename'. This file is opened in read mode using file pointer 'fp1'. Character 'c' is used to read characters
from the file and print them to the output. User is asked for the line number in the file to be deleted. The file
pointer is rewinded back and all the lines of the file except for the line to be deleted are copied into another file
"copy.c". Now "copy.c" is renamed to the original filename. The original file is opened in read mode and the
modified contents of the file are displayed on the screen.

17.Write a program to replace a specified line in a text file.

Program: Program to replace a specified line in a text file.

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

```
int main(void) {
FILE *fp1, *fp2;
//'filename'is a 40 character string to store filename
char filename[40];
char c;
int del_line, temp = 1;
//asks user for file name
printf("Enter file name: ");
//receives file name from user and stores in 'filename'
scanf("%s", filename);
fp1 = fopen(filename, "r");
//open file in read mode
c = getc(fp1);
//print the contents of file .
while (c != EOF) {
printf("%c", c);
c = getc(fp1);
}
//ask user for line number to be deleted.
printf(" \n Enter line number to be deleted and replaced");
scanf("%d", &del_line);
//take fp1 to start point.
rewind(fp1);
//open copy.c in write mode
fp2 = fopen("copy.c", "w");
c = getc(fp1);
while (c != EOF) {
if (c == '\n') {
temp++;
}
//till the line to be deleted comes,copy the content from one file to other
if (temp != del_line){
putc(c, fp2);
}
else //when the line to be deleted comes
{
while ((c = getc(fp1)) != '\n') {
}
//read and skip the line ask for new text
printf("Enter new text");
//flush the input stream
fflush(stdin);
putc('\n', fp2);
//put '\n' in new file
while ((c = getchar()) != '\n')
putc(c, fp2);
//take the data from user and place it in new file
fputs("\n", fp2);
temp++;
}
//continue this till EOF is encountered
c = getc(fp1);
}
//close both files
fclose(fp1);
fclose(fp2);
//remove original file
remove(filename);
//rename new file with old name opens the file in read mode
rename("copy.c", filename);
fp1 = fopen(filename, "r");
//reads the character from file
c = getc(fp1);
//until last character of file is encountered
while (c != EOF){
printf("%c", c);
//all characters are printed
c = getc(fp1);
}
//close the file pointer
fclose(fp1);
return 0;
}
```

Output:

```
Enter file name:abc.txt
hi.
hello
how are you?
hope the same
Enter line number of the line to be deleted and replaced:4
Enter new text: sayonara see you soon
hi.
hello
how are you?
sayonara see you soon
```

Explanation:

In this program, the user is asked to type the name of the file. The File by name entered by user is opened in read mode. The line number of the line to be replaced is asked as input. Next the data to be replaced is asked. A new file is opened in write mode named "copy.c". Now the contents of original file are transferred into new file and the line to be modified is deleted. New data is stored in its place and remaining lines of the original file are also transferred. The copied file with modified contents is replaced with the original file's name. Both the file pointers are closed and the original file is again opened in read mode and the contents of the original file is printed as output.

18.Write a program to find the number of lines in a text file.

Number of lines in a file can be determined by counting the number of new line characters present.

Program: Program to count number of lines in a file.

```
#include <stdio.h>
int main()
/* Ask for a filename and count number of lines in the file*/
{
//a pointer to a FILE structure
FILE *fp;
int no_lines = 0;
//consider 40 character string to store filename
char filename[40], sample_chr;
//asks user for file name
printf("Enter file name: ");
//receives file name from user and stores in a string named 'filename'
scanf("%s", filename);
//open file in read mode
fp = fopen(filename, "r");
//get character from file and store in sample_chr
sample_chr = getc(fp);
while (sample_chr != EOF) {
//Count whenever sample_chr is '\n'(new line) is encountered
if (sample_chr == '\n')
{
//increment variable 'no_lines' by 1
no_lines=no_lines+1;
}
//take next character from file.
sample_chr = getc(fp);
}
fclose(fp); //close file.
printf("There are %d lines in %s \n", no_lines, filename);
return 0;
}
```

Output:

```
Enter file name:abc.txt
There are 4 lines in abc.txt
```

Explanation:

In this program, name of the file to be read is taken as input. A file by the given name is opened in read-mode using a File pointer 'fp'. Characters from the file are read into a char variable 'sample_chr' with the help of getc function. If a new line character('\n') is encountered, the integer variable 'no_lines' is incremented. If the character read into 'sample_char' is not a new line character, next character is read from the file. This process is continued until the last character of the file(EOF) is encountered. The file pointer is then closed and the total number of lines is shown as output.

19.Write a C program which asks the user for a number between 1 to 9 and shows the number. If the user inputs a number out of the specified range, the program should show an error and prompt

the user for a valid input.

Program: Program for accepting a number in a given range.

```
#include<stdio.h>
int getnumber();
int main() {
int input = 0;
//call a function to input number from key board
input = getnumber();
//when input is not in the range of 1 to 9,print error message
while (!((input <= 9) && (input >= 1))) {
printf("[ERROR] The number you entered is out of range");
//input another number
input = getnumber();
}
//this function is repeated until a valid input is given by user.
printf("\nThe number you entered is %d", input);
return 0;
}/
/this function returns the number given by user
int getnumber() {
int number;
//asks user for a input in given range
printf("\nEnter a number between 1 to 9 \n");
scanf("%d", &number);
return (number);
}
```

Output:

```
Enter a number between 1 to 9
45
[ERROR] The number you entered is out of range
Enter a number between 1 to 9
4
The number you entered is 4
```

Explanation:

getfunction() function accepts input from user. 'while' loop checks whether the number falls within range or not
and accordingly either prints the number(If the number falls in desired range) or shows error message(number is
out of range).

20.Write a program to display the multiplication table of a given number.

Program: Multiplication table of a given number

```
#include <stdio.h>
int main() {
int num, i = 1;
printf("\n Enter any Number:");
scanf("%d", &num);
printf("Multiplication table of %d: \n", num);
while (i <= 10) {
printf("\n %d x %d = %d", num, i, num * i);
i++;
}
return 0;
}
```

Output:

```
Enter any Number:5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

Explanation:

We need to multiply the given number (i.e. the number for which we want the multiplication table)
with value of 'i' which increments from 1 to 10.

Find Us on FACEBOOK …If u like this document Please Like Our PAGE

Give FEED BACK in WWW.INDIAENOW.IN

End