ASSIGNMENT — 1

NAME OF STUDENT : Challa. Ravi teja
REGISTER NUMBER : 192325065
CLASS CODE : CSA0271
ASSIGNMENT NO : 01
COURSE NAME : C— programming.

1. FLOYD'S TRIANGLE:

Floyd's triangle in C is a right-angled triangular array of natural numbers.

Aim: To write the C- programm to print the Floyd's triangle of n rows using for loop.

Programm:

```
#include<stdio.h>
#include<conio.h>
int main ()
{
int n,i,c,a=1;
scanf("%d", &n);
printf("enter the no.of rows of floyd triangle to print:\n");
for(i=1;i<=n;i++)
{
for(c=1;c<=i;c++)
{
printf("%d",a);
a++;
}
printf("\n");
}
return 0;
}
```

Input: 05

output:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

2. **PASCAL TRIANGLE:**

Pascal triangle is a pattern similar to a triangle.

Aim: To write the C-programm to print the pascal triangle of n rows using for loop.

Programm:

```
#include<stdio.h>
#include<conio.h>
int main ()
{
int rows, coef=1, space, i, j;
printf("enter the number of rows:\n");
scanf("%d", &rows);
for (i=0; i<rows; i++)
{
for (space=1; space<=rows-i; space++)
print (" ");
for (j=0; j<=i; j++)
{
If (j==0||i==0)
coef = 1;
else
coef = coef *(i-j+1)/j;
printf("%4d", coef);
}
printf("\n");
}
return 0;
}
```

Input: Rows – 6

Output:

```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5   10  10  5   1
```

3. RHOMBUS STAR PATTERN:

Rhombus star pattern in C is a design of stars in a rhombus structure.

Aim: write a C-programm to print rhombus star pattern of N rows using for loop.

Programm:

```
#include<stdio.h>
int main ( )
{
int n;
printf("enter the number of rows: \n");
scanf("%d",&n);
for (int i=1;i<=n;i++) {
for (int j=1;j<=n-i;j++) {
printf(" ");
}
for (int j=1;j<=2*i-1;j++) {
printf("*");
}
printf("\n");
}
for(int i=n-1;j>=1;i--){
for(int j=1;j<=n-i;j++){
printf(" ");
}
for (int j=1;j<=2*i-1;j++){
printf("*");
}
printf("\n");
}
return 0;
}
```

Input: Rows = 04

Output:

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

4. DIAMOND STAR PATTERN:

Diamond star pattern in c is a design of stars in a Diamond structure.

Aim: write a c-programm to print diamond star pattern in N rows using for loop.

programm:

```
#include<stdio.h>
int main ( ) {
int n,c,k;
Printf("enter number of rows: \n");
Scanf("%d", &n);
for (k=1; k<=n; k++) {
for (c=1; c<=n-k; c++)
print(" ");
for (c=1; c<=2*k-1; c++)
printf("*");
printf("\n");
}
for (k=1; k<=n-1; k++) {
for (c=1; c<=k; c++)
printf(" ");
for (c=1; c<=2*(n-k)-1; c++)
printf("*");
printf("\n");
}
return 0;
}
```

Input: number of rows = 05

output:

5. STAR PATTERN :

Star pattern in c programm is a design of stars in right angled triangle.

Aim : Write a C - programm to print star pattern of N rows using for loop.

programm :

```
#include <stdio.h>
#include <conio.h>
int main ( )
{
int rows;
scanf("%d", &rows);
printf("enter the number of rows: \n");
for (int i=1; i<=rows ; i++)
{
for (int j=1; j<=i ; j++)
{
printf(" * ");
}
printf ("\n");
}
return 0;
}
```

Input :

Number of rows = 6

output :

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

6. PALINDROME :

palindrome is a word or number or other sequence of symbols that reads the same backwards and forwards.

Aim: To write the C - programm and find the integer is whether it is palindrome or not.

Programm :

```
# include <stdio.h>
# include <conio.h>
void main ( )
{
int n, m, sum=0;
scanf("%d", &n);
int temp=n;
while (n>0)
{
m=n%10;
sum=(sum*10)+m;
n=n/10;
}
If (sum==temp)
{
printf("palindrome");
}
else
{
printf("not a palindrome");
}
}
```

Input :

⇒ 363

output:

⇒ It is "palindrome".

7. Matrix multiplication:

In mathematics, particularly in linear algebra, matrix multiplication is a binary operation that produces a matrix from two matrices.

Aim: write a c-programm to print the multiplication of the two known matrices.

Programm:

```
#include<stdio.h>
void main ( ){
int a[10][10], b[10][10], mul[10][10], r, c, i, j, k;
Scanf("%d", &r);
Scanf("%d",&c);
for (i=0 ; i<r ;i++) {
for(j=0;j<c ; j++) {
Scanf("% d", &a[i][j]);
}
}
for(i=0;i<r;i++) {
for(j=0 ;j<c;j++){
Scanf("%d", &b[i][j]);
}
}
for(i=0; i<r ;i++) {
for(j=0; j<c;j++) {
mul[i][j]=0;
for(k=0; k<c; k++) {
MUL[i][j]+=a[i][k]* b[k][j];
}
}
}
for(i=0;i<r; i++) {
for(j=0;j<c;j++){
printf("%d\t", mul[i][j]);}
Printf("\n");
}
}
```

Input: 2 2 1234 2134

output:

$$\begin{vmatrix} 8 & 9 \\ 19 & 18 \end{vmatrix}$$

8. GCD AND LCM OF TWO INTEGERS:

In GCD and LCM of two integers. GCD stands for greatest common division and LCM stands for least common multiple.

Aim: write a c-programm to print the GCD and LCM of any two integers.

Programm:

```
#include<stdio.h>
int main ( )
{
int num1, num2, gcd, lcm, count = 1 , small;
scanf("%d%d", &num1, &num2);
printf("enter two integers\n");
small = (num1<num2) ? num1 : num2;
while (count<=small)
{
If (num1%count==0 && num2%count==0)
{
gcd = count;
}
count++;
}
lcm = (num1*num2)/gcd;
printf("GCD=%d\nLCM = %d\n", gcd, lcm);
return 0;
}
```

Input:

10

15

Output:

GCD = 5

LCM = 30

9. HCF OF TWO NUMBERS:

The HCF of two numbers. HCF stands for the highest common factor.

Aim: write a C-programm to print the highest common factor using any two integers.

Programm:

```
#include <stdio.h>
#include <conio.h>
int main ()
{
int a,b,i,hcf;
a=12;
b=15;
for (i=1; i<=a || i<=b; i++)
{
If (a%i==0 && b%i==0)
hcf=i;
}
printf("hcf=%d", hcf);
return 0;
}
```

Input:

12

15

output:

HCF of two number 12 and 15 is "3"

10. VOWELS AND CONSONANTS:

The vowels and consonants states that the count the number of vowels and consonants in a sentence.

Aim: To write the C-programm to print the count of vowels and consonants in the given sentence.

Programm:

```
#include<stdio.h>
#include<conio.h>
int main()
{
char str[100];
int i, vowels=0, consonants=0;
printf("enter a sentences:\n");
fgets(str, size of(str), stdin);
for(i=0; str[i]!='\0'; i++){
If(str[i]=='a'||str[i]=='e'||str[i]=='i'||str[i]=='o'||str[i]=='u'
  ||str[i]=='A'||str[i]=='E'||str[i]=='I'||str[i]=='O'||str[i]=='U')
{
vowels++;
}
else if((str[i]>='a'&&str[i]<='z')||(str[i]>='A'&&str[i]<='Z')
{
consonants++;
}
}
printf("The number of vowels is: %d\n", vowels);
printf("The number of constants is: %d\n", consonants);
return 0;
}
```

Input:

C-programm

Output:

vowels = 2

consonants = 7