

**Practice sheet1**

Do the following Practice programmes

**1. Write a program to display an average of 3 inputted numbers.**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    // declare variable
    float num1, num2, num3;
    float sum, avg; clrscr();

    // take inputs
    printf("Enter three Numbers: ");
    scanf("%f %f %f",&num1, &num2, &num3);

    // calculate sum
    sum = num1 + num2 + num3;

    // calculate average
    avg = sum / 3;

    // display entered numbers
    printf("Entered numbers are: %.2f, %.2f and %.2f\n",num1, num2, num3);

    // display sum and average
    printf("Sum=%.2f\n", sum);
    printf("Average=%.2f\n",avg );
    getch();
    return 0;
}
```

**/\* Output**

```
Enter three Numbers: 10 20 30
Entered numbers are: 10.00, 20.00
and 30.00
Sum=60.00
Average=20.00 */
```

**2. Write a program to calculate simple interest for a given principle amount, rate of interest and no of years.**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    float p,r,t,int_amt; clrscr();
    printf("Input principle, Rate of interest & time to find simple interest: \n");
    scanf("%f %f %f",&p,&r,&t);
    int_amt=(p*r*t)/100;
    printf("Simple interest = %f",int_amt);
    getch(); return 0;
}
```

**3. Write a program to display area of triangle.**

```
/* C program to find area of a triangle if base and height are given */
#include<stdio.h>
#include<conio.h>
int main()
{
    float base, height, area; clrscr();
```



```
/* Input base and height of triangle */
printf("Enter base of the triangle: ");
scanf("%f", &base);
printf("Enter height of the triangle: ");
scanf("%f", &height);
/* Calculate area of triangle */
area = (base * height) / 2;
/* Print the resultant area */
printf("Area of the triangle = %.2f sq. units", area);
getch(); return 0;
}
```

**4. Write a program to calculate area of circle.**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    float radius, area; clrscr();
    printf("\nEnter the radius of Circle : ");
    scanf("%d", &radius);
    area = 3.14 * radius * radius;
    printf("\nArea of Circle : %f", area);
    getch(); return 0;
}
```

**5. Write a program to calculate area of rectangle.**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    float length,breadth;
    float area; clrscr();
    printf(" Enter the Length of a Rectangle : ");
    scanf("%f",&length);
    printf("\n Enter the Breadth of a Rectangle : ");
    scanf("%f",&breadth);
    area = length * breadth;
    printf("\n Area of Rectangle is : %f",area);
    getch(); return 0;
}
```

**OUTPUT:**

```
Enter the Length of Rectangle : 3.2
Enter the Breadth of Rectangle : 4
Area of Rectangle : 12.800000
```

**6. Write a program to check whether inputted no is positive or negative.**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int num; clrscr();
    printf("Input a number :");
    scanf("%d", &num);
    if (num >= 0)
        printf("%d is a positive number \n", num);
    else
        printf("%d is a negative number \n", num);
}
```

**7. Write a program to check whether inputted no is divided by 9 or not?**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int num; clrscr();
    printf("Enter a number: ");
    scanf("%d", &num);
    if(num%9==0)
    {
        printf("%d is divisible by 9",num);
    }
    else
    {
        printf("%d is not divisible by 9",num);
    }
    getch(); return 0;
}
```

**8. Write a program to check whether given no is odd or even?**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int num; clrscr();
    printf("Enter an integer: ");
    scanf("%d", &num);
    // True if num is perfectly divisible by 2
    if(num % 2 == 0)
        printf("%d is even.", num);
    else
        printf("%d is odd.", num);
    getch(); return 0;
}
```

**9. Write a program to display Y<sup>th</sup> power of X.**

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    double x,num; clrscr();
    x=pow(5.0,2.0);
    printf("power is : %f",x);
    getch();
}
*****OR*****
/* Program to find exponential without using pow() method */
#include <stdio.h>
int main()
{
    int n, exp, exp1;
    long long int value = 1;           // %lld format specifier
    printf("\n Enter the number and its exponential:");
```



```
scanf("%d %d",&n, &exp);
exp1 = exp;    // storing original value for future use
while(exp > 0)  // same as while(--exp)!=-1)
{
    value *= n;  // multiply n to itself exp times
    exp--;
}
printf("\n\n %d^%d = %lld \n\n", n, exp1, value);
printf("\n Coding is Fun !!!");
return 0;
}
```

**OUTPUT:**

Enter the number and its  
exponential:5  
3  
5^3 = 125  
Coding is Fun !!!

**10. Write a program to check given inputted year is leap year or not.**

```
/* C Program to identify a Leap Year */
#include<stdio.h>
#include<conio.h>
int main()
{
    int year;
    printf("Enter the year to check if it is a leap year: ");
    scanf("%d", &year);
    if(year%400 == 0)    // divisible by 4, 100 and 400
    {
        printf("\n\n %d is a leap year\n", year);
    }
    else if(year%100 == 0) // divisible by 100 and 4 and not divisible by 400
    {
        printf("\n\n %d is not a leap year\n", year);
    }
    else if(year%4 == 0)  // divisible only by 4 and not by 100
    { // %0 nd is used to represent the number in n digits with leading 0's
        printf("\n\n %07d is a leap year\n", year);
    }
    else // not divisible by 4
    { printf("\n\n %d is not a leap year \n", year); }

    printf("\n\t Coding is Fun !!!!!"); getch(); return 0;
}
```

**11. Write a program which accept temperature in Celsius and convert into Fahrenheit.  $F = (C * (9/5)) + 32$**

```
#include <stdio.h>
int main()
{ float celsius, fahrenheit;
    printf("Please Enter temperature in Celsius: \n");
    scanf("%f", &celsius);
    // Convert the temperature from celsius to fahrenheit
    fahrenheit = ((celsius * 9)/5) + 32;

    // fahrenheit = ((9/5) * celsius) + 32;

    // fahrenheit = ((1.8 * celsius) + 32;

    printf("\n %.2f Celsius = %.2f Fahrenheit", celsius, fahrenheit);
    return 0;
}
```

**12. Write a program which accept Fahrenheit and convert into Celsius.**

$$C = (F - 32) * 5 / 9$$

```
//C program to convert Fahrenheit to Celsius
```



```
#include <stdio.h>
int main()
{
    float celsius,fahrenheit;    // Reads temperature in Fahrenheit
    printf("\nEnter temperature in Fahrenheit:");
    scanf("%f",&fahrenheit);
    // Fahrenheit to Celsius conversion formula
    celsius=(fahrenheit - 32)* 5/9;
    // Print the result
    printf("\n Celsius = %.3f",celsius);
    return 0;
}
```

**13. Write a program to display minimum(smallest) number from three given numbers.**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int num1,num2,num3; clrscr();
    printf("Enter three numbers:");
    scanf("%d %d %d", &num1, &num2, &num3);
    if( (num1 < num2) && (num1 < num3))
    {
        printf("%d is smallest",num1);
    }
    else if(num2 < num3)
    {
        printf("%d is smallest",num2);
    }
    else
    {
        printf("%d is smallest",num3);
    }
    getch(); return 0;
}
```

**14. Write a program to display maximum(greatest) number from three given numbers.**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int x, y, z; clrscr();
    printf(" Enter three Integer numbers for x, y, z : ");
    scanf("%d %d %d", &x, &y, &z);
    if((x > y) && (x > z) )
        printf("\n x is greatest : %d",x);
    else if(y > z)
        printf("\n y is greatest : %d",y);
    else
        printf("\n z is greatest : %d", z);
    getch(); return 0;
}
```

**15. Write a program to display reverse number. E.g. I/P =>1234 O/P => 4321**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int n, reverse=0, rem; clrscr();
    printf("Enter a number: ");
    scanf("%d", &n);
    while(n!=0)
    {
        rem=n%10;
        reverse=reverse*10 + rem;
        n=n/10;
    }
    printf("Reversed Number: %d", reverse);
    getch(); return 0;
}
```

/\*Output:

Enter a number: 1234

Reversed Number: 4321 \*/

**16. Write a program to check whether the entered alphabet is vowel or consonant.**

```
#include<stdio.h>
#include<conio.h>
int main()
{ char ch; clrscr();
  printf("Enter any Character : ");
  scanf("%c", &ch);
  switch(ch)
  {
      case 'a':
      case 'A':
      case 'e':
      case 'E':
      case 'i':
      case 'I':
      case 'o':
      case 'O':
      case 'u':
      case 'U':
          printf("\n %c is a Vowel.\n\n", ch);
          break;
      default:
          printf("\n %c is Consonant. \n", ch);
  }
  getch(); return 0;
}
```

**17. Write a program to print first 10 natural numbers and its sum and average.**

```
#include<stdio.h>
#include<conio.h>
void main()
{ int j, sum = 0;
  float avg=0.0; clrscr();
  printf("The first 10 natural number is :\n");
  for (j = 1; j <= 10; j++)
```



```
{
    sum = sum + j;
    printf("%d ",j);
}
avg=sum/10;
printf("\nThe Sum is : %d\n", sum);
printf("\nThe Average is : %f\n", avg);
getch();
}
```

**18. Write a Program to count the Number of Digits. E.g. I/P =>1234 O/P => 4**

```
#include<stdio.h>
#include<conio.h>
void main()
{ long long n;
  int count = 0; clrscr();
  printf("Enter an integer: ");
  scanf("%lld", &n);
  // iterate until n becomes 0 and remove last digit from n in each iteration
  // increase count by 1 in each iteration
  while (n != 0)
  {
      n /= 10;          // n = n/10
      ++count;
  }
  printf("Number of digits: %d", count);
  getch();
}
```

**/\* Output:**

Enter an integer:1234  
Number of digits: 4 \*/

**19. Write a program to print factorial of n.**

```
#include<stdio.h>
#include<conio.h>
int main()
{ int a=1,num;
  long int fact=1; clrscr();

  printf("Enter a number: ");
  scanf("%d",&num);
  while(a <= num)
  {
      fact=fact*a;
      a++;
  }
  printf("\n Factorial of %d is: %ld", num, fact);
  getch(); return 0;
}
```

**OUTPUT:**

Enter a number: 6  
Factorial of 6 is: 720

**20. Write a program to calculate the sum of digits of the entered number.**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int n,sum=0,rem;
    printf("Enter a number:");
    scanf("%d",&n);
    while(n>0)
```

**Output:**

Enter a number:654  
Sum is=15



```
{
    rem=n%10;
    sum=sum+rem;
    n=n/10;
}
printf("Sum is=%d", sum);
return 0;
}
```

**21. Write a program to check whether the number is palindrome or not.**

```
#include <stdio.h>
int main()
{
    int n, reversedN = 0, remainder, originalN;
    printf("Enter an integer: ");
    scanf("%d", &n);
    originalN = n;
    // reversed integer is stored in reversedN
    while (n != 0)
    {
        remainder = n % 10;
        reversedN = reversedN * 10 + remainder;
        n /= 10;
    }
    // palindrome if originalN and reversedN are equal
    if (originalN == reversedN)
        printf("%d is a palindrome.", originalN);
    else
        printf("%d is not a palindrome.", originalN);
    return 0;
}
```

**OUTPUT:**

Enter an integer: 1001

1001 is a palindrome.

Enter an integer: 12521

12521 is a palindrome.

**22. Write a program to check whether the entered number is Prime or not.**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int n, i, flag = 0; clrscr();
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    for(i = 2; i <= n / 2; ++i)
    {
        // condition for non-prime
        if (n % i == 0)
        {
            flag = 1;
            break;
        }
    }
    if (n == 1)
    {
        printf("1 is neither prime nor composite.");
    }
    else
    {
        if (flag == 0)
            printf("%d is a prime number.", n);
        else
            printf("%d is not a prime number.", n);
    }
}
```

**OUTPUT:**

Enter a positive integer: 17

17 is a prime number.

Enter a positive integer: 28

28 is not a prime number.





```
        printf("%d is not a prime number.", n);  
    }  
    getch();    return 0;  
}
```

**23. Write a program to Fibonacci series of n numbers.**

```
#include<stdio.h>  
#include<conio.h>  
int main()  
{  
    int i, n, t1 = 0, t2 = 1, nextTerm; clrscr();  
    printf("\n Enter the number of terms: ");  
    scanf("%d", &n);  
    printf("\n Fibonacci Series: ");  
    for (i = 1; i <= n; ++i)  
    {  
        printf("%d, ", t1);  
        nextTerm = t1 + t2;  
        t1 = t2;  
        t2 = nextTerm;  
    }  
    getch();    return 0;  
}
```

Enter the number of terms: 10  
Fibonacci Series: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

**24. Write a program to check whether the entered number is Armstrong or not.**

```
#include<stdio.h>  
#include<conio.h>  
int main()  
{  
    int num, originalNum, remainder, result = 0; clrscr();  
    printf("Enter a three-digit integer: ");  
    scanf("%d", &num);  
    originalNum = num;  
  
    while(originalNum != 0)  
    {  
        // remainder contains the last digit  
        remainder = originalNum % 10;  
        result += remainder * remainder * remainder;  
        // removing last digit from the original number  
        originalNum /= 10;  
    }  
    if (result == num)  
        printf("%d is an Armstrong number.", num);  
    else  
        printf("%d is not an Armstrong number.", num);  
    getch();    return 0;  
}
```

**OUTPUT:**

Enter a three-digit integer: 371  
371 is an Armstrong number.

**25. Write a program to count characters in the entered string.**

```
#include<stdio.h>  
#include<conio.h>  
#include<string.h>  
int main()  
{  
    char str[100];  
    int count = 0; clrscr();
```



```
printf("Enter String :");
gets(str);

//Counts each character except space
for(int i = 0; i < strlen(str); i++)
{
    if(str[i] != ' ')
        count++;
}
//Displays the total number of characters present in the given string
printf("Total number of characters in a string: %d", count);
getch();    return 0;
}
```

**26. Write a program to create choice based menu for calculator operations.**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int num1,num2,opt;
    long int res;    clrscr();
    printf("Enter the first Integer :");
    scanf("%d", &num1);
    printf("Enter the second Integer :");
    scanf("%d", &num2);

    printf("\n1-Addition.\n2-Substraction.\n3-Multiplication.\n4-Division.\n5-Power.\n6-
Exit.\n");
    printf("\n Input your option :");
    scanf("%d", &opt);
    switch(opt)
    {
        case 1:
            printf("The Addition of %d and %d is: %d\n",num1,num2,num1+num2);
            break;
        case 2:
            printf("The Substraction of %d and %d is: %d\n",num1,num2,num1-num2);
            break;
        case 3:
            printf("The Multiplication of %d and %d is: %d\n",num1,num2,num1*num2);
            break;
        case 4:
            if(num2==0)
            {
                printf("The second integer is zero. Devide by zero.\n");
            }
            else
            {
                printf("The Division of %d and %d is : %d\n",num1,num2,num1/num2);
            }
            break;
        case 5:
            res=pow(num1,num2);
            printf("The Power of %d ^ %d is : %ld\n",num1,num2,res);
            break;
        case 6:
            break;
    }
```



```

default:
    printf("\n Input correct option....\n");
    break;
}
printf("\n End of Calculator...\n");
getch();
}
/* OUTPUT:
Enter the first Integer :12
Enter the second Integer :2
1-Addition.
2-Substraction.
3-Multiplication.
4-Division.
5-Power.
6-Exit.
Input your option :5
The Power of 12 ^ 2 is : 144

End of Calculator... */

```

**27. Write a program to accept N numbers and print in reverse order using array.**

```

#include<stdio.h>
#include<conio.h>
#define MAX_SIZE 100          // Defines maximum size of array
int main()
{
    int arr[MAX_SIZE];
    int size, i; clrscr();
    printf("Enter size of the array: "); /* Input size of array */
    scanf("%d", &size);
    printf("Enter elements in array: "); /* Input array elements */
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }
    printf("\nArray in reverse order: ");
    for(i = size-1; i>=0; i--)
    {
        printf("%d\t", arr[i]);
    }
    getch(); return 0;
}

```

**Input:**

Input array elements: 10, 5, 16, 35, 500

**Output:**

Array elements after reverse: 500, 35, 16, 5, 10

**28. Write a program to display maximum and minimum element from the entered array.**

```

#include <stdio.h>
int main()
{
    int a[5], i, max, min;
    for(i=0; i<5; i++)
    {
        printf("\n Enter Array element :");
        scanf("%d", &a[i]);
    }
    max=min=a[0];
    for(i=0; i<5; i++)
    {

```



```

        if(a[i] > max)
        {
            max = a[i];
        }
        if(a[i] < min)
        {
            min = a[i];
        }
    }
    printf("\n The Given Array Element is:");
    for(i=0;i<5;i++)
    {
        printf(" %d ",a[i]);
    }
    printf("\n From The Array Element Largest Number is: %d", max);
    printf("\n From The Array Element Smallest Number is: %d", min);
    return 0;
}
/* Output:
Enter Array element :34
Enter Array element :55
Enter Array element :3
Enter Array element :5
Enter Array element :2
The Given Array Element is: 34 55 3 5 2
From The Array Element Largest Number is: 55
From The Array Element Smallest Number is: 2 */

```

**29. Write a program to arrange elements of the entered array in ascending order.**

```

#include<stdio.h>
#include<conio.h>
int main()
{
    int count, temp, i, j, number[50]; clrscr();

    printf("How many numbers are u going to enter ? ");
    scanf("%d", &count);
    for(i=0;i<count;i++)
    {
        printf("Enter %d numbers: ",i+1);
        scanf("%d", &number[i]);
    }
    //This is the main logic of bubble sort algorithm
    for(i=0 ;i < count-1 ; i++)
    {
        for(j=0; j < count-i-1 ; j++)
        {
            if(number[j] < number[j+1]) // ascending order    // for descending <
            {
                temp=number[j];
                number[j]=number[j+1];
                number[j+1]=temp;
            }
        }
    }
    printf("Sorted elements in ascending order using Bubble sort:\n ");
}

```



```
for(i=0;i<count;i++)
    printf(" %d", number[i]);

getch();    return 0;
}
```

/\* **OUTPUT:**

How many numbers are u going to enter ? 6

Enter 1 numbers: -2

Enter 2 numbers: 0

Enter 3 numbers: 34

Enter 4 numbers: 5

Enter 5 numbers: 78

Enter 6 numbers: 23

Sorted elements: -2 0 5 23 34 78 \*/

**30. Write a program to generate mark sheet of 5 Student and display appropriate result as follow:**

Percentage	Result
<b>&gt;=70</b>	<b>Distinction</b>
<b>&gt;=60</b>	<b>First Class</b>
<b>&gt;=50</b>	<b>Second Class</b>
<b>&gt;=35</b>	<b>Pass Class</b>
<b>&lt;35</b>	<b>Fail</b>

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
    int rollno;
    char name[50],grade[20];
    float sub1,sub2,sub3,sub4,sub5,total,per;           clrscr();

    printf("\nEnter Roll No :");
    scanf("%d",&rollno);
    printf("\nEnter Name :");
    //gets(name);           // enter full name   or   %[^\n]s
    scanf("%s",name);       //single word

    printf("\nEnter the Marks of CS : ");
    scanf("%f", &sub1);
    printf("\nEnter the Marks of DMA : ");
    scanf("%f", &sub2);
    printf("\nEnter the Marks of CPPM : ");
    scanf("%f", &sub3);
    printf("\nEnter the Marks of Math : ");
    scanf("%f", &sub4);
    printf("\nEnter the Marks of IC : ");
    scanf("%f", &sub5);
    total = sub1 + sub2 + sub3 + sub4 + sub5;
    per = total / 5;

    if(per >= 70)           // 70 to 100
        strcpy(grade, "Distinction");
    else if(per>=60 )       // 60 to 69
```



```

        strcpy(grade, "First Class");
    else if(per>=50 )    // 50 to 59
        strcpy(grade, "Second Class");
    else if(per>=35 )    // 35 to 49
        strcpy(grade, "Pass Class");
    else
        strcpy(grade, "Fail");
    printf("\n Student Roll No is : %d",rollNo);
    printf("\n Student Name is : %s\n\n",name);
    printf("Subject  Max.Marks  Obt.Marks  \n");
    printf("-----\n");
    printf("CS      100      %.2f      \n",sub1);
    printf("DMA      100      %.2f      \n",sub2);
    printf("CPPM      100      %.2f      \n",sub3);
    printf("Math      100      %.2f      \n",sub4);
    printf("IC       100      %.2f      \n",sub5);
    printf("-----\n");

    printf("Total Marks %.2f | Percentage %.2f %% \n", total, per);
    printf("\n Grade = %s you got %s." , name, grade);
    getch();    return 0;
}

```

/\* **OUTPUT:**

```

Enter Roll No :46
Enter Name :Ganesh
Enter the Marks of CS : 45
Enter the Marks of DMA : 86
Enter the Marks of CPPM : 89
Enter the Marks of Math : 25
Enter the Marks of IC : 63
Student Roll No is : 46
Student Name is : Ganesh
Subject  Max.Marks  Obt.Marks

```

```

-----
CS      100      45.00
DMA      100      86.00
CPPM      100      89.00
Math      100      25.00
IC       100      63.00
-----

```

```

Total Marks 308.00 | Percentage 61.60 %

```

```

Grade = Ganesh you got First Class.    */

```

**31. Write a C program to toggle each character in a string.**

```

#include <stdio.h>
int main()
{
    char  str[100];
    int   counter;
    printf("Enter a string: ");
    gets(str);

```



```
// toggle each string characters
for(counter=0; str[counter] !=NULL; counter++)
{
    if(str[counter]>='A' && str[counter]<='Z')
        str[counter]=str[counter]+32;    //convert into lower case
    else if(str[counter]>='a' && str[counter]<='z')
        str[counter]=str[counter]-32;    //convert into upper case
    }
printf("String after toggle each characters: %s",str);
return 0;
}
/* OUTPUT :
Enter a string: This is a Test String.
String after toggle each characters: tHIS IS A tEST sTRING. */
```

**32. Write a C program to find sum of first and last digit of any number.**

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int num, sum=0, firstDigit, lastDigit;    clrscr();
    /* Input a number from user */
    printf("Enter any number to find sum of first and last digit: ");
    scanf("%d", &num);

    /* Find last digit to sum */
    lastDigit = num % 10;

    /* Copy num to first digit */
    firstDigit = num;

    /* Find the first digit by dividing num by 10 until first digit is left */
    while(num >= 10)
    {
        num = num / 10;
    }
    firstDigit = num;
    sum = firstDigit + lastDigit;    /* Find sum of first and last digit*/
    printf("Sum of first and last digit = %d", sum);
    getch();    return 0;
}
/* Output:
Input number: 12345
Sum of first and last digit: 6 */
```

**33. Write a C program to convert Decimal to binary.**

```
#include <stdio.h>
int main()
{
    int n, c, k;
    printf("Enter an integer in decimal number system: ");
    scanf("%d", &n);
```



```
printf("%d in binary number system is:\n", n);
for (c = 31; c >= 0; c--)
{
    k = n >> c;

    if (k & 1)
        printf("1");
    else
        printf("0");
}
printf("\n");    return 0;
}
/* OUTPUT:
Enter an integer in decimal number system: 63
in binary number system is:
0000000000000000000000000111111    */
```

**34. Write a C program to convert Decimal to Octal.**

```
#include<stdio.h>
int main()
{
    int decimalnum, octalnum = 0, temp = 1;

    printf("Enter a Decimal Number: ");
    scanf("%d", &decimalnum);

    while(decimalnum != 0)
    {
        octalnum = octalnum + (decimalnum % 8) * temp;
        decimalnum = decimalnum / 8;
        temp = temp * 10;
    }
    printf("Equivalent Octal Number: %d", octalnum);
    return 0;
}
```

**OUTPUT:**

Enter a Decimal Number: 436  
Equivalent Octal Number: 664

**35. Write a C program to Convert Decimal to Hexadecimal.**

```
#include <stdio.h>
int main()
{
    long decimalnum, quotient, remainder;
    int i, j = 0;
    char hexadecimalnum[100];
    printf("\n Enter decimal number: ");
    scanf("%ld", &decimalnum);
    quotient = decimalnum;
    while (quotient != 0)
    {
        remainder = quotient % 16;
        if (remainder < 10)
            hexadecimalnum[j++] = 48 + remainder;
        else
            hexadecimalnum[j++] = 55 + remainder;
        quotient = quotient / 16;
    }
}
```





```
// display integer into character
printf("\n Hexadecimal value is :");
for (i = j; i >= 0; i--)
    printf("%c", hexadecimalnum[i]);
return 0;
}
/* OUTPUT:
Enter decimal number: 123
Hexadecimal value is :7B  */
```

### Unit 5-Pointer Examples

#### **/\* C Program to add Two Number Using Pointer \*/**

```
#include<stdio.h>
int main()
{
    int a, b, c;
    int *pa, *pb, *pc;
    pa=&a, pb=&b, pc=&c;
    printf("Enter two number: ");
    scanf("%d %d", pa, pb);
    // pa and pb have address of a & b respectively
    *pc = *pa + *pb;
    printf("Sum=%d\n",*pc);
    return 0;
}
```

**OUTPUT:**

Enter two number: 10,20  
10 + 20 = 30

#### **/\* C Program to find Area of Circle Using Pointer \*/**

```
#include<stdio.h>
#define PI 3.14
int main()
{
    double r, area;
    double *pr, *parea;
    pr= &r, parea=&area;

    printf("Enter radius: ");
    scanf("%lf",pr);

    *parea = PI * (*pr) * (*pr);
    printf("Area of radius %.3lf = %.3lf\n", *pr, *parea);
    return 0;
}
```

**OUTPUT:**

Enter radius: 9.7  
Area of radius 9.700000 = 295.44



## **Pattern Program Practice**

### **1. Pattern programs in C for Floyd's Triangle**

```
#include<stdio.h>
int main()
{
    int n, r, c, a=1;
    printf("Enter number of rows: ");
    scanf("%d",&n);

    for(r=1; r<=n; r++)
    {
        for(c=1; c<=r; c++)
        {
            printf("%5d",a++);
        }

        printf("\n");
    }
    return 0;
}
```

Enter number of rows: 5

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

### **2. Mix Pattern programs in C**

```
#include<stdio.h>
int main()
{
    int n, r, c;
    printf("Enter number of rows: ");
    scanf("%d",&n);
    for(r=1; r<=n; r++)
    {
        for(c=1; c<=r; c++)
        {
            if((r+c)%2==0)
                printf("* ");
            else
                printf("$ ");
        }
        printf("\n");
    }
    return 0;
}
```

Enter the number of rows: 5

```
*
$ *
* $ *
$ * $ *
* $ * $ *
```

### **3. Pattern programs in C**

```
#include<stdio.h>
int main()
{
    int n, r, c, a=0;
    char ch='A';
    printf("Enter number of rows: ");
    scanf("%d",&n);
    for(r=1; r<=n; r++)
    {
        for(c=1; c<=r; c++)
```

Enter the number of rows: 5

```
A
1 1
B B B
2 2 2
C C C C C
```



```
{
    if(r%2==1)
        printf("%5c",ch);
    else
        printf("%5d",a);
}
printf("\n");
if(r%2==0)
    ch++;
else
    a++;
}
return 0;
}
```

#### 4. Full Pyramid of Numbers

```
#include <stdio.h>
int main()
{
    int i, space, rows, k = 0, count = 0, count1 = 0;
    printf("Enter the number of rows: ");
    scanf("%d", &rows);
    for (i = 1; i <= rows; ++i)
    {
        for (space = 1; space <= rows - i; ++space)
        {
            printf(" ");
            ++count;
        }
        while (k != 2 * i - 1)
        {
            if (count <= rows - 1)
            {
                printf("%d ", i + k);
                ++count;
            }
            else
            {
                ++count1;
                printf("%d ", (i + k - 2 * count1));
            }
            ++k;
        }
        count1 = count = k = 0;
        printf("\n");
    }
    return 0;
}
```

Enter the number of rows: 5

```

      1
     2 3 2
    3 4 5 4 3
   4 5 6 7 6 5 4
  5 6 7 8 9 8 7 6 5
```

#### 5. Pascal's Triangle Program in C

```
#include <stdio.h>
int main()
{
    int rows, coef = 1, space, i, j;
```



```
printf("Enter the number of rows: ");
scanf("%d", &rows);
for (i = 0; i < rows; i++)
{
    for (space = 1; space <= rows - i; space++)
        printf(" ");
    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;
}
```

Enter the number of rows: 5

```

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

## 6. Inverted full pyramid of \* Program in C

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

Enter the number of rows: 5

```

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

## 7. Full Pyramid of \* program in C

```
#include <stdio.h>
int main()
{
    int i, space, rows, k = 0;
    printf("Enter the number of rows: ");
    scanf("%d", &rows);
    for (i = 1; i <= rows; ++i, k = 0)
    {
        for (space = 1; space <= rows - i; ++space)
        {
            printf(" ");
        }
        while (k != 2 * i - 1)
```

Enter the number of rows: 5

```

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



```
{
    printf("* ");
    ++k;
}
printf("\n");
}
return 0;
}
```

### 8. Full Pyramid with Numbers program in C

```
#include<stdio.h>
int main()
{
    int n, r, c, k, a;
    printf("Enter number of rows: ");
    scanf("%d",&n);
    for(r=1;r<=n;r++)
    {
        for(c=1;c<=n-r;c++)
            printf(" ");

        for(k=1;k<=(2*r-1);k++)
        {
            if(k<r)
                printf("%d",k);
            else if(k==r)
            {
                printf("%d",k);
                a=k;
            }
            else
                printf("%d",--a);
        }
        printf("\n");
    } // end of Outer loop
    return 0;
}
```

Enter number of rows: 5

```
1
121
12321
1234321
123454321
```

### 9. Print Following Pattern in C

```
#include<stdio.h>
int main()
{
    int n;
    char ch;
    printf("Enter number of lines: ");
    scanf("%d",&n);
    for(int i=0; i<=n; i++)
    {
        ch = 'A';
        for(int j=0; j<=n-i; j++, ch++)
        {
            printf("%c",ch);
        }
    }
}
```

Enter number of lines: 5

```
ABCDEFGFEDCBA
ABCDE    EDCBA
ABCD      DCBA
ABC        CBA
AB          BA
A            A
```



```
}  
if(i==0)  
{  
    printf("%c",ch);  
}  
else  
{  
    for(int k=0; k<(2*i)+1; k++)  
    {  
        printf(" ");  
    }  
}  
ch--;  
for(int j=0; j<=n-i; j++, ch--)  
{  
    printf("%c",ch);  
}  
printf("\n");  
}  
return 0;  
}
```

### **UNIT – 1 – NOTES**

**Write note on basic data types in 'C'. How can we do implicit and explicit conversion ? when it is needed ? OR Explain Type Conversion. Also describe the condition in which explicit and implicit conversion is used.**

Data types specify how we enter data into our programs and what type of data we enter. C language has some predefined set of data types to handle various kinds of data that we can use in our program. These datatypes have different storage capacities.

C language supports 2 different type of data types:

#### **Primary data types:**

These are fundamental data types in C namely integer(int), floating point(float), character(char) and void.

**Derived data types:**

Derived data types are nothing but primary datatypes but a little twisted or grouped together like array, structure, union and pointers.(Read Unit 1 in detail)

**What is Typecasting in C?**

Typecasting is converting one data type into another one. It is also called as data conversion or type conversion. It is one of the important concepts introduced in 'C' programming.

'C' programming provides two types of type casting operations:

- 1 Implicit type casting
- 2 Explicit type casting

**Implicit type casting**

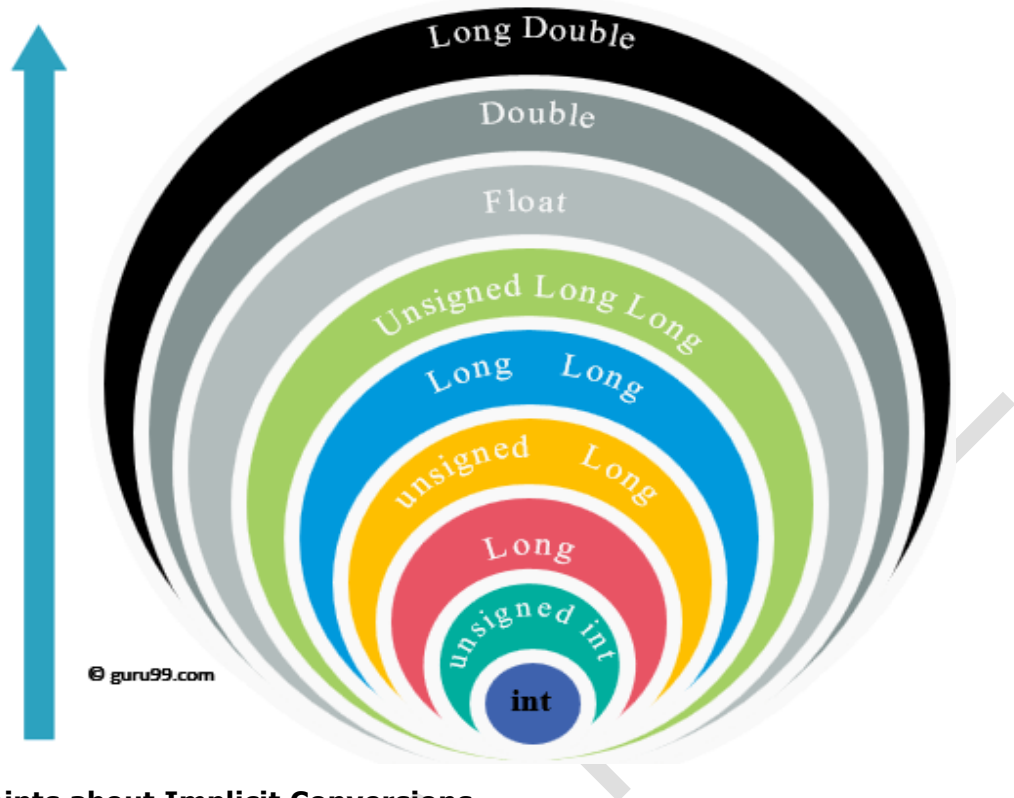
Implicit type casting means conversion of data types without losing its original meaning. This type of typecasting is essential when you want to change data types without changing the significance of the values stored inside the variable.

Implicit type conversion happens automatically when a value is copied to its compatible data type. During conversion, strict rules for type conversion are applied. If the operands are of two different data types, then an operand having lower data type is automatically converted into a higher data type. This type of type conversion can be seen in the following example.

```
#include<stdio.h>
int main(){
    short a=10; //initializing variable of short data type
    int b; //declaring int variable
    b=a; //implicit type casting
    printf("%d\n",a);
    printf("%d\n",b);
}
/* output:
10
10 */
```

**Arithmetic Conversion Hierarchy**

The compiler first proceeds with promoting a character to an integer. If the operands still have different data types, then they are converted to the highest data type that appears in the following hierarchy chart:



### Important Points about Implicit Conversions

- 1 Implicit type of type conversion is also called as standard type conversion. We do not require any keyword or special statements in implicit type casting.
- 2 Converting from smaller data type into larger data type is also called as type promotion. In the above example, we can also say that the value of s is promoted to type integer.
- 3 The implicit type conversion always happens with the compatible data types.

### Explicit type casting

In implicit type conversion, the data type is converted automatically. There are some scenarios in which we may have to force type conversion. Suppose we have a variable div that stores the division of two operands which are declared as an int data type.

```
int result, var1=10, var2=3;

result=var1/var2;
```

In this case, after the division performed on variables var1 and var2 the result stored in the variable "result" will be in an integer format. Whenever this happens, the value stored in the variable "result" loses its meaning because it does not consider the fraction part which is normally obtained in the division of two numbers.

To force the type conversion in such situations, we use explicit type casting.

It requires a type casting operator. The general syntax for type casting operations is as follows:

```
(type-name) expression
```

Here,

The type name is the standard 'C' language data type.

An expression can be a constant, a variable or an actual expression.





**Write a program to demonstrate implementation of explicit type-casting in 'C'.**

```
#include<stdio.h>
int main()
{
    float a = 1.2;
    //int b = a;           //Compiler will throw an error for this
    int b = (int)a + 1;
    printf("Value of a is %f\n", a);
    printf("Value of b is %d\n",b);
    return 0;
}
```

**OUTPUT:**

Value of a is 1.200000  
Value of b is 2

## ASCII Table

Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char	Dec	Hex	Oct	Binary	Char
0	00	000	0000000	NUL (null character)	32	20	040	0100000	space	64	40	100	1000000	@	96	60	140	1100000	`
1	01	001	0000001	SOH (start of header)	33	21	041	0100001	!	65	41	101	1000001	A	97	61	141	1100001	a
2	02	002	0000010	STX (start of text)	34	22	042	0100010	"	66	42	102	1000010	B	98	62	142	1100010	b
3	03	003	0000011	ETX (end of text)	35	23	043	0100011	#	67	43	103	1000011	C	99	63	143	1100011	c
4	04	004	0000100	EOT (end of transmission)	36	24	044	0100100	\$	68	44	104	1000100	D	100	64	144	1100100	d
5	05	005	0000101	ENQ (enquiry)	37	25	045	0100101	%	69	45	105	1000101	E	101	65	145	1100101	e
6	06	006	0000110	ACK (acknowledge)	38	26	046	0100110	&	70	46	106	1000110	F	102	66	146	1100110	f
7	07	007	0000111	BEL (bell (ring))	39	27	047	0100111	'	71	47	107	1000111	G	103	67	147	1100111	g
8	08	010	0001000	BS (backspace)	40	28	050	0101000	(	72	48	110	1001000	H	104	68	150	1101000	h
9	09	011	0001001	HT (horizontal tab)	41	29	051	0101001	)	73	49	111	1001001	I	105	69	151	1101001	i
10	0A	012	0001010	LF (line feed)	42	2A	052	0101010	*	74	4A	112	1001010	J	106	6A	152	1101010	j
11	0B	013	0001011	VT (vertical tab)	43	2B	053	0101011	+	75	4B	113	1001011	K	107	6B	153	1101011	k
12	0C	014	0001100	FF (form feed)	44	2C	054	0101100	,	76	4C	114	1001100	L	108	6C	154	1101100	l
13	0D	015	0001101	CR (carriage return)	45	2D	055	0101101	-	77	4D	115	1001101	M	109	6D	155	1101101	m
14	0E	016	0001110	SO (shift out)	46	2E	056	0101110	.	78	4E	116	1001110	N	110	6E	156	1101110	n
15	0F	017	0001111	SI (shift in)	47	2F	057	0101111	/	79	4F	117	1001111	O	111	6F	157	1101111	o
16	10	020	0010000	DLE (data link escape)	48	30	060	0110000	0	80	50	120	1010000	P	112	70	160	1110000	p
17	11	021	0010001	DC1 (device control 1)	49	31	061	0110001	1	81	51	121	1010001	Q	113	71	161	1110001	q
18	12	022	0010010	DC2 (device control 2)	50	32	062	0110010	2	82	52	122	1010010	R	114	72	162	1110010	r
19	13	023	0010011	DC3 (device control 3)	51	33	063	0110011	3	83	53	123	1010011	S	115	73	163	1110011	s
20	14	024	0010100	DC4 (device control 4)	52	34	064	0110100	4	84	54	124	1010100	T	116	74	164	1110100	t
21	15	025	0010101	NAK (negative acknowledge)	53	35	065	0110101	5	85	55	125	1010101	U	117	75	165	1110101	u
22	16	026	0010110	SYN (synchronize)	54	36	066	0110110	6	86	56	126	1010110	V	118	76	166	1110110	v
23	17	027	0010111	ETB (end transmission block)	55	37	067	0110111	7	87	57	127	1010111	W	119	77	167	1110111	w
24	18	030	0011000	CAN (cancel)	56	38	070	0111000	8	88	58	130	1011000	X	120	78	170	1111000	x
25	19	031	0011001	EM (end of medium)	57	39	071	0111001	9	89	59	131	1011001	Y	121	79	171	1111001	y
26	1A	032	0011010	SUB (substitute)	58	3A	072	0111010	:	90	5A	132	1011010	Z	122	7A	172	1111010	z
27	1B	033	0011011	ESC (escape)	59	3B	073	0111011	;	91	5B	133	1011011	[	123	7B	173	1111011	{
28	1C	034	0011100	FS (file separator)	60	3C	074	0111100	<	92	5C	134	1011100	\	124	7C	174	1111100	
29	1D	035	0011101	GS (group separator)	61	3D	075	0111101	=	93	5D	135	1011101	]	125	7D	175	1111101	}
30	1E	036	0011110	RS (record separator)	62	3E	076	0111110	>	94	5E	136	1011110	^	126	7E	176	1111110	~
31	1F	037	0011111	US (unit separator)	63	3F	077	0111111	?	95	5F	137	1011111	_	127	7F	177	1111111	DEL