Week-1 : Statements , Expressions & Conditionals

Aim –1 : write a c program to print the memory allocation required for all the datatypes in C language

Program:

```
#include<stdio.h>
 main() {
    int intType;
    float floatType;
    double doubleType;
    char charType;


    printf("Size of int: %zu bytes\n", sizeof(intType));
    printf("Size of float: %zu bytes\n", sizeof(floatType));
    printf("Size of double: %zu bytes\n", sizeof(doubleType));
    printf("Size of char: %zu byte\n", sizeof(charType));

    return 0;
}
```

Output: Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
Size of char: 1 byte

Aim-- 2 : write a c program to print the given number is even or odd.

```
#include <stdio.h>
main() {
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);
```

```
    if(num % 2 == 0)
        printf("%d is even.", num);
    else
        printf("%d is odd.", num);


}
```

Output: Enter an integer: 9
9 is odd.

Enter an integer: -4
-4 is even.

Aim—3 : Write C a Program to find the given number is Positive ,negative or zero.

```
#include <stdio.h>

main()
{
    int num;

    printf("Enter the number : ");
    scanf("%d", &num);

    if (num > 0)
        printf("%d is positive.", num);
    else if (num < 0)
        printf("%d is negative.", num);
    else if (num == 0)
        printf("%d is zero.", num);

    return 0;
```

```
}
```

Input: Enter the number = 2
Output: 2 is positive

Input: Enter the number = -554
Output: -554 is negative

Input: Enter the number = 0
Output: 0 is zero

Aim—4 : a) Write C a Program to find the swap 2 numbers using 3rd variable

```
#include<stdio.h>
 main() {
      double first, second, temp;
      printf("Enter first number: ");
      scanf("%lf", &first);
      printf("Enter second number: ");
      scanf("%lf", &second);


      temp = first;


      first = second;


    second = temp;

      printf("\nAfter swapping, firstNumber = %.2lf\n", first);
      printf("After swapping, secondNumber = %.2lf", second);
      return 0;
}
```

```
Enter first number: 1.20
Enter second number: 2.45

After swapping, firstNumber = 2.45
After swapping, secondNumber = 1.20
```

b. Write C a Program to find the swap 2 numbers without using 3rd variable

```
#include <stdio.h>
int main() {
    double a, b;
    printf("Enter a: ");
    scanf("%lf", &a);
    printf("Enter b: ");
    scanf("%lf", &b);


    a = a - b;


b = a + b;


a = b - a;

    printf("After swapping, a = %.2lf\n", a);
    printf("After swapping, b = %.2lf", b);
    return 0;
}
Enter a: 10.25
Enter b: -12.5
After swapping, a = -12.50
After swapping, b = 10.25
```

Aim—4 Write a C program to check number is perfect square or not.

```
#include <stdio.h>
#include <math.h>
int main()
{
int num;
int iVar;
float fVar;
printf("Enter an integer number: ");
scanf("%d",&num);
fVar=sqrt(num);
iVar=fVar;
if(iVar==fVar)
printf("%d is a perfect square.",num);
else
printf("%d is not a perfect square.",num);
return 0;
}
```

Output:

```
Enter an integer number: 64
64 is a perfect square
Enter an integer number: 23
23 is not a perfect square
```

Aim--5:

```
#include<stdio.h>

#include<conio.h>

int main()

{

   int a,b;

   int op;
```

```
printf(" 1.Addition\n 2.Subtraction\n 3.Multiplication\n 4.Division\n");
printf("Enter the values of a & b: ");
scanf("%d %d",&a,&b);
printf("Enter your Choice : ");
scanf("%d",&op);
switch(op)
{
case 1      :
   printf("Sum of %d and %d is : %d",a,b,a+b);
   break;
case 2      :
   printf("Difference of %d and %d is : %d",a,b,a-b);
   break;
case 3      :
   printf("Multiplication of %d and %d is : %d",a,b,a*b);
   break;
case 4      :
   printf("Division of Two Numbers is %d : ",a/b);
   break;
default     :
   printf(" Enter Your Correct Choice.");
   break;
}
```

```
    return 0;
}
```

Output for Program:

```
1.Addition
2.Subtraction
3.Multiplication
4.Division
Enter the values of a & b:  20 15
Enter your Choice : 1
Sum of 20 and 15 is : 35
```

Week-2

1. write a C program to print all the factors of a given number

```
#include <stdio.h>

main()
{
   int i, n;
   printf("Enter any number to find its factor: ");
   scanf("%d", &n);
   printf("All factors of %d are: ", n);
   for(i=1; i<=n; i++)
   {
      if(n % i == 0)
      {
         printf("%d, ",i);
      }
   }
   return 0;
```

```
}
```

Input/output:

```
Enter any number to find its factor: 9
All factors of 9 are:        1, 3, 9,
Enter any number to find its factor: 10
All factors of 10 are:       1,2,5,10
```

Explination: Input number from user. Store it in some variable say n
Run a loop from 1 to n, increment 1 in each iteration. The loop structure should look like for(i=1; i<=n; i++).
For each iteration inside loop check current counter loop variable i is a factor of n or not. To check factor we check divisibility of number by performing modulo division i.e. if(n % i == 0) then i is a factor of n. If i is a factor of n then print the value of i.

2. write a C program to find the factorial of a given number

```
#include<stdio.h>
main()
{
  int i,factorial=1,n;
  printf("Enter a number: ");
  scanf("%d",&n);
  for(i=1;i<=n;i++)
     factorial=factorial*i;
  printf("Factorial of %d is: %d",n,factorial);
  return 0;
}
```

Inpur/output:

```
Enter a number:  5
Factorial of 5 is: 120
Enter a number: 10
Factorial of 10 is: 3628800
```

3. write a C program to find whether a given is number palindrome or not

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

```
int main() {
   int n, rev = 0, rem, temp;
   printf("Enter an integer: ");
   scanf("%d", &n);
   temp = n;

   while (n != 0) {
      rem = n % 10;
      rev= (rev * 10 )+rem;
      n /= 10;
   }
 if (temp == rev)
      printf("%d is a palindrome.", rev);
   else
      printf("%d is not a palindrome.", rev);

   return 0;
}
```

input/output:

Enter an integer: 123

123 is not a palindrome.

Enter an integer: 121

121 is a palindrome.

4. write a C program to print Fibonacci upto given ‘n’ number of terms

```
#include <stdio.h>
int main() {
   int i, n, a = 0, b = 1, c;
   printf("Enter the number of terms: ");
   scanf("%d", &n);
   do
   {
                i++;
                printf("%d, ", a);
      c = a + b;
      a = b;
```

```
        b = c;

                }

    while(i <= n);
          }
```

input/output:

Enter the number of terms: 5

0, 1, 1, 2, 3, 5,

Enter the number of terms: 7

0, 1, 1, 2, 3, 5,8,13

5. write a C program to find whether a given is number prime or not

```
#include <stdio.h>
 main()
 {
   int n, i, flag = 0;
   printf("Enter a positive integer: ");
   scanf("%d", &n);
   for (i = 2; i <= n / 2; ++i) {
      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);
   }

   return 0;
```

}

input/output:

Enter a positive integer: 5

5 is a prime number.

Enter a positive integer: 20

20 is not a prime number.

Week –3 :

1. write a c program to print first n prime numbers .

```
#include<stdio.h>
 main()
{
   printf("\n\n\t\t RGUKT - Best place to learn\n\n\n");
   int n,i = 3, count, c;
   printf("\nEnter the number of prime numbers required :  ");
   scanf("%d", &n);
   if(n >= 1)
   {
      printf("\n\nFirst %d prime numbers are :  ", n);
      printf("2 ");
   }
   for(count = 2; count <= n; i++)
   {
      for(c = 2; c < i; c++)
      {
```

```
            if(i%c == 0)
                break;
        }
        if(c == i)
        {
            printf("%d ", i);
            count++;
                        }
    }
    return 0;
}
```

Input/Output:

RGUKT - Best place to learn

Enter the number of prime numbers required : 23

First 23 prime numbers are : 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83

Enter the number of rows : 6

2. Write a C program to print pascal's triangle

```
#include<stdio.h>
main()
{
int r, c = 1, space, i, j;
printf("\nEnter the number of rows : ");
scanf("%d",&r);
printf("\n");
```

```
for(i=0; i<r; i++)
{
for(space=1; space <= r-i; space++)
printf(" ");

for(j=0; j <= i; j++)
{
if (j==0 || i==0)
c = 1;
else
c = c*(i-j+1)/j;

printf("%4d", c);
}
printf("\n\n");
}
return 0;
}
```

Input/output:

Enter the number of rows : 6

```
     1


   1   1


 1   2   1


1   3   3   1


1   4   6   4   1
```

1 5 f 10 10 5 1

3. Write a C program to print the 1st ‘n’ perfect number for a given ‘n’ value

```
#include <stdio.h>
int main()
{
   int i, j, upper, sum;
   printf("Enter upper limit: ");
   scanf("%d", &upper);
   printf("All Perfect numbers between 1 to %d:\n", upper);
   for(i=1; i<=upper; i++)
   {
      sum = 0;
      for(j=1; j<i; j++)
      {
         if(i % j == 0)
         {
            sum += j;
         }
      }
      if(sum == i)
      {
         printf("%d, ", i);
```

```
        }
    }
    return 0;
}
```

Input/output:

```
Enter upper limit: 50
All Perfect numbers between 1 to 50:  6, 28,
```

4.

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

Input/output:

```
Enter the number of rows: 6
```

*

* *

* * *

* * * *

* * * * *

* * * * * *

5. Write a C program to print the pattern

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

```
    for (c = 1; c <= r; r++)
      printf(" ");
    for (c = 1 ; c <= 2*(n-r)-1; c++)
      printf("*");

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

Input/output:

```
Enter number of rows:6
     *
    ***
   *****
  *******
 *********
***********
 *********
  *******
   *****
    ***
     *
```

6. write a C program to print the following pattern

Enter the number of rows: 5

5 5 5 5 5

4 4 4 4

3 3 3

2 2

1

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

WEEK-4:

1. write a c program to take an input array of n numbers and find sum , product, mean of n values

```
#include<stdio.h>
int main()
{
   int n, sum = 0,i,product=1,mean=0, array[100];
   printf("Enter the number of integers you want to add: ");
   scanf("%d", &n);
   printf("\n\nEnter %d integers \n\n", n);
   for(i = 0; i < n; i++)
   {
      scanf("%d", &array[i]);
      sum += array[i];
      mean=sum/n;
      product*=array[i];
        }
   printf("\nsum = %d\n\n", sum);
   printf("\nmean = %d\n\n", mean);
        printf("\nproduct = %d\n\n", product);
   return 0;
}
```

Inpu/output:

Enter the number of integers you want to add: 5

Enter 5 integers

20 30 40 50 10

sum = 150

mean = 30

product = 12000000

2. Write a C program to find Second Largest and smallest in given array

```
#include <stdio.h>
 main ()
   {
  int num[30],i, j, a, n, count, avg;

       printf("Enter the value of N:");
       scanf("%d", &n);
       printf("Enter the numbers :");
       for (i = 0; i < n; ++i)
          scanf("%d", &num[i]);
       for (i = 0; i < n; ++i)
       {
          for (j = i + 1; j < n; ++j)
```

```
    {
        if (num[i] < num[j])
        {
            a = num[i];
            num[i] = num[j];
            num[j] = a;
        }
    }
}
printf("The numbers arranged in descending order are given below \n");
for (i = 0; i < n; ++i)
{
    printf("%d\n", num[i]);
}
printf("The 2nd largest number is  = %d\n", num[1]);
printf("The 2nd smallest number is = %d\n", num[n - 2]);

avg = (num[1] + num[n - 2]) / 2;
count = 0;
for (i = 0; i < n; ++i)
{
    if (avg == num[i])
    {
```

```
            ++count;
        }
    }

    if (count == 0 )
        printf("The average of %d  and %d is = %d is not in the array \n",
        num[1], num[n - 2], avg);
    else
        printf("The average of %d  and %d in array is %d in numbers \n",
        num[1], num[n - 2], count);
}
```

Input/output:

Enter the value of N:5

Enter the numbers :12 34 23 3 1

The numbers arranged in descending order are given below

34

23

12

3

1

The 2nd largest number is = 23

The 2nd smallest number is = 3

The average of 23 and 3 is = 13 is not in the array

Week-5

1. Write a program to find the addition and substation for the given two matrices of size M X N and P X Q respectively

```
#include<stdio.h>

int main()

{

   int n, m, i, j, f[10][10], s[10][10], sum[10][10], diff[10][10];

   printf("\nEnter the number of rows and columns of the first matrix \n\n");

   scanf("%d%d", &m, &n);

   printf("\nEnter the %d elements of the first matrix \n\n", m*n);

   for(i = 0; i < m; i++)   // to iterate the rows

      for(j = 0; j < n; j++)   // to iterate the columns

         scanf("%d", &f[i][j]);

   printf("\nEnter the %d elements of the second matrix \n\n", m*n);

   for(i = 0; i < m; i++)

      for(j = 0; j < n; j++)

         scanf("%d", &s[i][j]);

   printf("\n\nThe first matrix is: \n\n");

   for(i = 0; i < m; i++)

   {

      for(j = 0; j < n; j++)

                        {

         printf("%d\t", f[i][j]);
```

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

printf("\n\nThe second matrix is: \n\n");
for(i = 0; i < m; i++)
{
    for(j = 0; j < n; j++)
    {
        printf("%d\t", s[i][j]);
    }
printf("\n");
}
for(i = 0; i < m; i++)
    for(j= 0; j < n; j++)
        sum[i][j] = f[i][j] + s[i][j];

printf("\n\nThe sum of the two entered matrices is: \n\n");
for(i = 0; i < m; i++)
{
    for(j = 0; j < n; j++)
    {
        printf("%d\t", sum[i][j]);
```

```
        }
        printf("\n");
    }
    for(i = 0; i < m; i++)
        for(j = 0; j < n; j++)
            diff[i][j] = f[i][j] - s[i][j];

    printf("\n\nThe difference(subtraction) of the two entered matrices is: \n\n");
    for(i = 0; i < m; i++)
    {
        for(j = 0; j< n; j++)
        {
            printf("%d\t", diff[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Input /output:

Enter the 6 elements of the second matrix

4 2 1 7 8 0

The first matrix is:

1 2 3

5 6 9

The second matrix is:

4 2 1

7 8 0

The sum of the two entered matrices is:

5 4 4

12 14 9

The difference(subtraction) of the two entered matrices is:

-3 0 2

-2 -2 9

2. write a c program to calculate matrix multiplication

```
#include<stdio.h>
main(){
int a[10][10],b[10][10],mul[10][10],r1,c1,r2,c2,i,j,k;
printf("enter the first number of row and column:");
scanf("%d%d",&r1,&c1);
printf("enter the first matrix element=\n");
for(i=0;i<r1;i++)
{
for(j=0;j<c1;j++)
{
scanf("%d",&a[i][j]);
}
}
printf("enter the first number of row and column:");
scanf("%d%d",&r2,&c2);
if(r2!=c1)
{
        printf("sorry");
}
else
{
printf("enter the second matrix element=\n");
```

```
for(i=0;i<r2;i++)
{
for(j=0;j<c2;j++)
{
scanf("%d",&b[i][j]);
}
}

printf("multiply of the matrix=\n");
for(i=0;i<r1;i++)
{
for(j=0;j<c2;j++)
{
mul[i][j]=0;
for(k=0;k<c1;k++)
{
mul[i][j]+=a[i][k]*b[k][j];
}
}
}
for(i=0;i<r1;i++)
{
for(j=0;j<c2;j++)
```

```
{
printf("%d\t",mul[i][j]);
}
printf("\n");
}
return 0;
}
}
```

Input /output:

1. enter the first number of row and column:2 3

enter the first matrix element=

1 1 1 1 1 1

enter the first number of row and column:3 2

enter the second matrix element=

2 3 4 5 6 7

multiply of the matrix=

12 15

12 15

2. enter the first number of row and column:2 3

enter the first matrix element=

1 2 3 4 5 6

enter the first number of row and column:2 3

sorry

3. Write a C program to print transpose of give matrix

```
#include <stdio.h>

int main()
{
  int m, n, i,j, mat[10][10], transpose[10][10];

  printf("Enter the number of rows and columns of a matrix\n");
  scanf("%d%d", &m, &n);
  printf("Enter elements of the matrix\n");

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

  for (i = 0; i < m; i++)
    for (j = 0; j < n; j++)
      transpose[j][i] = mat[i][j];

  printf("Transpose of the matrix:\n");

  for (i = 0; i < n; i++) {
    for (j = 0; j < m; j++)
```

```
      printf("%d\t", transpose[i][j]);
    printf("\n");
  }

  return 0;
}
```

Input/output:

Enter the number of rows and columns of a matrix

3 2

Enter elements of the matrix

1 2 3 4 2 6

Transpose of the matrix:

1 3 2

2 4 6