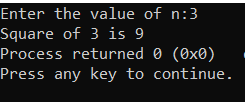
**Tutorial 10**

**Q1.** #include <stdio.h>

int sq(int n);

int main()

{ int N,square;

printf("Enter the value of n:");

scanf("%d",&N);

square=sq(N);

printf("Square of %d is %d",N,square);

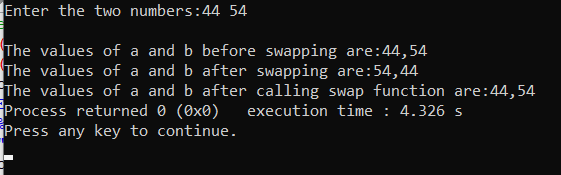
return 0; }

int sq(int n)

{

return n\*n;

}

**Q2. Call by value:-**

#include<stdio.h>

int swap(int,int);

int main()

{ int a,b;

printf("Enter the two numbers:");

scanf("%d%d",&a,&b);

printf("\nThe values of a and b before swapping are:%d,%d",a,b);

swap(a,b);

printf("\nThe values of a and b after calling swap function are:%d,%d",a,b); }

int swap(int a,int b)

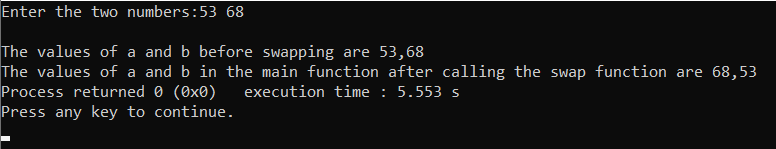
{ int temp;

temp=a;

a=b;

b=temp;

printf("\nThe values of a and b after swapping are:%d,%d",a,b); }

**Call by reference:-**

#include <stdio.h>

int swap(int \*,int \*);

int main()

{ int a,b;

printf("Enter the two numbers:");

scanf("%d%d",&a,&b);

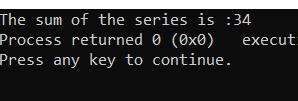
printf("\nThe values of a and b before swapping are %d,%d",a,b);

swap(&a,&b);

printf("\nThe values of a and b in the main function after calling the swap function are %d,%d",a,b); }

int swap(int \*a,int \*b)

{ int temp; temp=\*a; \*a=\*b; \*b=temp; }

 **Q3.** #include <stdio.h>

int fact(int);

int main()

{ int sum;

sum=fact(1)/1+fact(2)/2+fact(3)/3+fact(4)/4+fact(5)/5;

printf("The sum of the series is :%d",sum); }

int fact(int n)

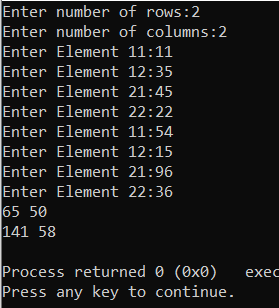
{ int N=0,f=1;

while(N<=n-1)

{ f=f+f\*N; N++; }

return f;

}

**Q4.** #include <stdio.h>

#include <conio.h>

void read(int a[10][10],int r,int c)

{ int i,j;

for(i=1;i<=r;i++)

{ for(j=1;j<=c;j++)

{ printf("Enter Element %d%d:",i,j);

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

}

}

void add(int m1[10][10],int m2[10][10],int m3[10][10],int r,int c)

{ int i,j;

for(i=1;i<=r;i++)

{ for(j=1;j<=c;j++)

{ m3[i][j]=(m1[i][j]+m2[i][j]); }

}

} void print(int m[10][10],int r,int c)

{ int i,j;

for(i=1;i<=r;i++)

{ for(j=1;j<=c;j++)

{ printf("%d",m[i][j]); }

printf("\n"); }

}

int main()

{ int m1[10][10],m2[10][10],m3[10][10],r,c;

printf("Enter number of rows:");

scanf("%d",&r);

printf("Enter number of columns:");

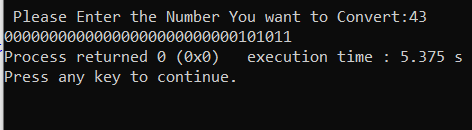
scanf("%d",&c);

read(m1,r,c);

read(m2,r,c);

add(m1,m2,m3,r,c);

print(m3,r,c); }



**Q5**. #include <stdio.h>

int convert(int n)

{ int j;

for(int i=31;i>=0;i--)

{ j=n>>i;

if(j&1)

printf("1");

else printf("0");

}

}

int main() { int n;

printf("\n Please Enter the Number You want to Convert:");

scanf("%d",&n);

convert(n);

return 0; }

**Q6.** #include <stdio.h>

#include <conio.h>

max(int [],int);

void main()

{

int a[]={10,5,45,12,19};

int n=5,m;

m=max(a,n);

printf("\nMAXIMUM NUMBER IS %d",m);

getch();

}

max(int x[],int k)

{

int t,i;

t=x[0];

for(i=1;i<k;i++)

{

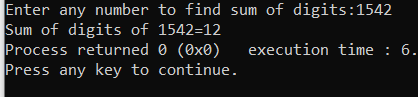
if(x[i]>t)

t=x[i];

}

return(t);

}

 **Q7.** #include

int summ(int num);

int main()

{ int num,sum;

printf("Enter any number to find sum of digits:");

scanf("%d",&num);

sum=summ(num);

printf("Sum of digits of %d=%d", num, sum);

return 0; }

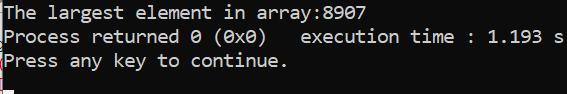
int summ(int num)

{ if(num==0)

return 0;

return ((num%10) + summ(num/10));

}

 **Q8.**  #include <stdio.h>

int largest(int list[],int l,int u);

int main()

{ int ar[5]={1,34,56,8907,53};

printf("The largest element in array:%d",largest(ar,0,4));

return 0;

}

int largest(int list[],int l,int u)

{ int max;

if (l==u)

return list[l];

else { max=largest(list,l+1,u);

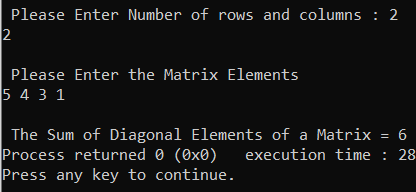
if (list[l]>=max)

return list[l];

else return

max; }

} ;

**Q9:** #include <stdio.h>

int main()

{ int i, j, rows, columns, a[10][10], Sum = 0;

printf("\n Please Enter Number of rows and columns : ");

scanf("%d %d", &i, &j);

printf("\n Please Enter the Matrix Elements \n");

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

{

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

{ scanf("%d", &a[rows][columns]); }

}

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

{ Sum = Sum + a[rows][rows]; }

printf("\n The Sum of Diagonal Elements of a Matrix = %d", Sum );

return 0; }

**Q10**:

**1.** x = 1, y = 2, z = 3

x = 10, y = 20.000000, z = 3

x = 10, y = 20.000000, z = 100

**2.** Error showing variable ‘var’ undeclared and no #include.

**3.** 4 is output

**4**. 83