# * Programs *

① Write 'C' Program for performing operations on array insertion & deletion

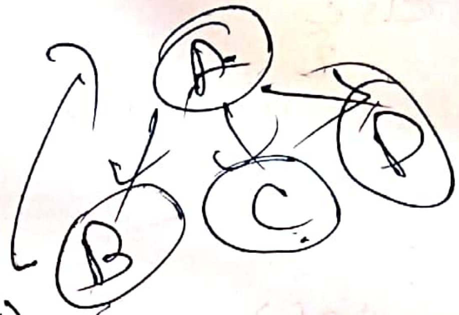
OR

Write 'C' Program for deletion of element of array.

OR

Write program to implement insert element in an array.

⇒
```
#include <stdio.h>
#include <conio.h>
void main()
{
 int i, n, index, a[30];
 printf("Enter size of array ");
 scanf("%d", &n);
 printf("\nEnter elements in array");
 for(i=0; i<n; i++)
  {
   printf(" a[%d] ", i);
   scanf("%d", &a[i]);
  }
 printf("\n Enter element to be deleted index: ");
 scanf("%d", &index);
 ~~if(index >= n+1)~~ if(index<0|| index >=n)
   {
   printf("Deletion not possible");
   }
 else
   {
    for(i=index; i<n-1; i++)
      {
       a[i] = a[i+1];
      }
   printf("Array after deletion is: ");
    for(i=0; i<n-1; i++)
      {
      printf("%d", a[i]);
      }
 getch();
}
```



Imp

② Factorial using recursion (6m)

```
#include<stdio.h>
long factorial(int n)
  {
    if(n<=1)
      {
      return(1);
      }
    else
      {
      return(n*factorial(n-1);
      }.
  }
void main()
  {
  long n;
  printf("Enter n ");
  Scanf("%d ",&n);
  printf("n!=%d ",factorial(n) );
}
```

Fibonacci

```
#include<stdio.h>
int fibonacci(int n)
{
   if(n<=1)
    {
    return n;
    }
   else
    {
   return fibonacci(n-1) + fibonacci(n-2);
 } }
int main()
   { int i,&n
   printf("enter n"); scan("%d",&n);
pf( " series");
```

```
for(i=0;i<n;i++)
{
print("%d", fibnacci(i));
}
return 0;
}
```

## Linear search

```
#include <stdio.h>
void main()
{
 int a[50], n, item, i;
 printf("Enter size of array");
 scanf("%d", &n);
 printf("Enter Elements");
 for(i=0; i<n; i++)
   {
   scanf("%d"&a[i]);
   }
 printf("~~Array~~ Enter item to be searched");
  Scanf("%d", &item);
  for(i=0; i<n; i++)
    {
      if(a[i]==item
        {
        printf("Element found at index=%d", i);
        break;
        }
    }
 if(i==n)
  {
   printf("Element not found");
  }
}
```

1 - Binary Search

```
#include<stdio.h>
void main()
 {
  int a[50],n;
  int low=0 ~~; int m~~, mid, high;
  printf("Enter size of array \n");
  Scanf("%d", &n);
  printf~~f~~("Enter elements");
  for(~~i=0~~ int i=0; i<n; i++)
      {
      scanf ("%d", &a[i]);
      }
   high=n-1;
   int item;
  printf("Enter searching item");
  Scanf("%d", &item);
   while (low<=High)
     {
     mid=(low+high)/2;
     if(a[mid]==item)
        {
       printf(" Item found at index %d", mid);
       return;
        }
      if(a[mid]<item)
        {
        low=mid+1;
        }
       else
        {
        high=mid-1;
        }
     }
   printf("not found");
 }
```

Bubble

#in

# Sorting

## Bubblesort.

```
#include<stdio.h>
void main()
{
 int a[50],i,j,n,temp;
 printf("Enter size of array");
 Scanf("%d",&n);
 printf("Enter elements");
 for(i=0;i<n;i++)
    {
     scanf("%d",&a[i]);
    }
 for(i=0; i<n-1;i++)
     {
       for(j=1 ;j<n-i-1 ;j++)
         {
            if(a[j] > a[j+1])
              {
               temp=a[j];
               a[j] =a[j+1];
               a[j+1]=temp;
              }
         }
     }
 printf("Sorted array");
 for(i=0;i<n;i++)
   {
    printf("%d",a[i]);

   }
}
```

# Selection Sort

```
#include<stdio.h>
 int main()
{
  int n;
  printf("Enter size of array");
  scanf("%d", &n);
  int ~~arr~~ a[50];
  printf("Enter elements");
  for(int i ; i<n ;i++).
      {
      ~~printf~~ scanf("%d", &a[i]);
      }
  for(i= 0;i<n-1 ;i++)
     {
      for( j= i+1 ;j<n ;j++)
          {
             if(a[j] <a[i])
              {  temp= a[j] ;
                  a[j]= a[i];
                   a[i] = temp;
              }
          }
     }
  pf("Sorted ");
  for(i=0; i<n; i++)
   {
     pf ("%d",a[i]);
   }
```

~~ider~~ Insertion sort

```
#include<stdio.h>
int main()
 {
 int i, j, n, key;
pf("Enter size of array");
scanf("%d", &n);
printf("Enter Elements");
for(i=0; i<n; i++)
    {
    scanf("%d", &a[i]);
    }
 for( i=1; i<n; i++)
    {
    key=a[i];
   for(j= i-1; j>=0 && a[j]>key; j++)
      {
      a [j+1]= ~~a~~ a[j];
      }
    a[j+1]=key;
  }
 pf ( Sorted array)
   for(i=0; i<n; i++)
      {
      pf ("%d", a[i]);
      }
}return 0;
```

Scanned with CamScanner

## Push pop

```
#include<stdio.h>
#define ~~MAX_SIZE~~ n 50.
int stack[100],top=-1;
void push();
void pop();
void display();
void main()
{
  int choice,n;
  printf("Enter size of array");
  ~~scanf("%d",&n);~~
   scanf("%d",&stack[n]);

 do{
  printf("\n1.push \n2.pop \n 3.Display");
  printf("Enter your choice");
  scanf("%d",&choice);
 switch(choice)
  {
   case 1: push();
           break;
   case 2: pop();
           break;
   case 3: display();
            break;
    default: printf("wrong choice");
   }
 }
 while(choice!=4);
 return 0;
}
```

void
2.

```
void push()
{
 if (top >= MAX_SIZE - 1)
  {
   printf("Overflow");
  }
else
 {
   int x;
   printf("Enter Element");
   scanf("%d", &x);
   stack[++top] = x;
 }
}
void pop()
{
  if (top <= -1)
   {
   printf("Underflow");
   }
else
  {
  printf("Popped element :%d", stack[top--]);
  }
}
void display()
{  if(top>=0)
    {
     printf("Element are );
    for (int i =top; i>=0; i--)
      {
       printf("%d\n", stack[i]);
      }
else
[illegible]          No elements
```

# Enqueue / Dequeue

```
#include<stdio.h>
#define max 5
int x, a[max], front=-1, rear=-1;
void insert()
{
 printf("Enter the element);
 scanf("%d", &x);
 if(front==max-1)
  {
  printf("Queue is overflow");
  }
 else
  front=0;
  rear=rear+1;
}  a[rear]=x;
void delete()
{
  if(front==-1)
  {
       underflow.
  }
 else
 printf("Poped %d", a[front]);
 front++;
 if(front>rear)
  {
  front=rear-1;
  }
}
```

```
void display()
  {
for (int i = front ; i<=rear ; i++)
     {
        a[i];

     }

main
     while(1)
Switch   {     ~
```