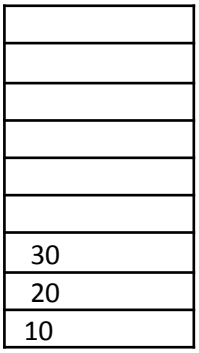
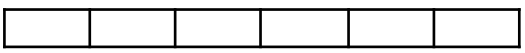
// Program to perform operations on array

```
#include<stdio.h>
Void   Read(int  [ ]);
Void   Disp(int  [ ]);
Void   Insert_pos(int  [ ] ) ;
Void   Delete_pos(int  [ ]);
Void Insert_by_order(int  [ ] ) ;
Void   Delete_bY_element(int  [ ]);

Int size = 10,   NE =0;

Void main( )
{
    Int   A[ size], choice;
   for( ; ;)
{
     Printf("enter choice:\n 1:Read\n2:Disp\n3:Insert\n4:Delete\5:Insert by order\n6:Delete by element\n7:exit\n");
     Scanf("%d", &choice);
       Switch(choice)
        {
```



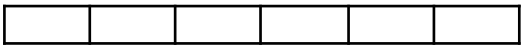

```
            case 1 :  Read(A);    break;
            case 2 :   Disp(A) ; break;
            case 3  : Insert_pos(A);  Disp(A);    break;
            case 4:    Delete_pos(A);  Disp(A);       break;
```

```
            case 5 :  Insert_by_order(A ) ; Disp(A);       break;
            case 6:  Delete_bY_element(A); Disp(A);       break;
           default : exit(0);
      }                                                    A = 1002 =X &A[0]
}
} // main

Void Read( int  X[  ]  )
 {
     Int  N, i  ;
    Printf("enter no of elements to read");
   Scanf("%d", &N);
   Printf("enter %d elements");
  For(i=0;i<N;i++)   scanf("%d", &X[ i]);
  NE = N
 }

Void Disp(int X[ ])
{  int i;
     If(NE ==0)   printf("empty");
    Return;
    Printf("array is:\n");
     For(i=0;  i< NE ; i++)   printf("%d  ", X[i]);
}
 Void Insert_pos ( int  X[ ])
{
   int   pos,  ele;
```



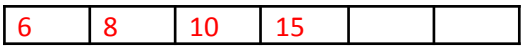
```
  If( NE == size )
    printf(" Array Full");   return;
 printf("enter position");       Scanf("%d", &pos);
```


```
printf("enter element");
```



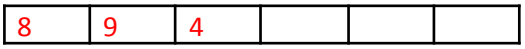
```
 Scanf("%d", &ele);
```

```
   If(pos >= 1 && pos<= NE+1)
{
      for(i=NE ;  i>= pos;   i--)
           X[ i ] = X[i-1];
      X[i ]=  ele ;
   NE++ ;
}
else    printf("Invalid position") ;
}


Void  Delete_pos ( int X[ ] )
  {   int   pos;
       // chk for empty
```




```
        If(NE==0)
         {     printf("array is empty");
           return;
           }
      Printf("enter position of the element to be deleted");
      Scanf("%d", &pos);
```

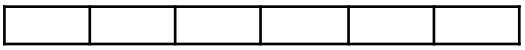

```
      // check for valid pos
```



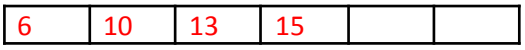

```
    If( pos >=1 && pos<=NE)
  {   // printf('deleted :%d\n", X[pos-1]);
      For(i= pos-1; i< NE-1; i++)
       {
              X[  i ] = X[ i+1  ] ;
```

```
        }
        NE--;
 }
  Else
 Printf("Invalid position");
}
```

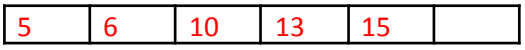
```
  Void Insert_by_order(int  X[ ] )
{
    int     ele,i;
```



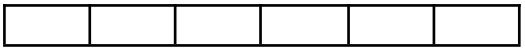
```
   If( NE == size )
     printf(" Array Full");   return;
```




```
printf("enter element");
```




```
 Scanf("%d", &ele);  // 5
```



```
        i=  NE-1;
while(    i>= 0  &&  X[i]  >ele    )
         {     X[i+1] =X[i];
               i--;
         }
        X[i+1] = ele;
         NE++ ;
}
```

```
Void   Delete_bY_element(int  X[ ])
 {
     {  int   ele;
      // chk for empty
       If(NE==0)
       {     printf("array is empty");
         return;
         }
     Printf("enter the element to be deleted");
     Scanf("%d", &ele); // 8
```

| 6 | 8 | 10 | 15 | | |
|---|---|---|---|---|---|

```
        For(i=0;  i<NE && X[I ]!=ele ;  i++);
         If( i==NE)
          {  printf("element not found in the list\n");
             return;
          }
           Printf("%d is deleted from position %d\n", X[i], i+1);
```

| 8 | 9 | 4 | 5 | 10 | |
|---|---|---|---|---|---|

```
        for( ;  i<NE-1   ; i++)
```

| 9 | 4 | 5 | 10 | | |
|---|---|---|---|---|---|

```
      {
              X[  i ] = X[ i+1  ] ;
     }
     NE--;
}
```