```
#include<stdio.h>
#include<stdlib.h>
int size;
int ne=0;
int *arr;
void read()
{
    int n;
    printf("enter the number of elements\n");
    scanf("%d",&n);
    printf("enter the array elements\n");
    for(int i=0;i<n;i++)
    scanf("%d",(arr+i));
    ne=n;
}
void insertbyorder(int key)
{
    int i;
    if(ne==size)
    {
        printf("array is full!\n");
        return;
    }
    for(i=ne-1;*(arr+i)>key;i--){
    *(arr+i+1)=*(arr+i);
    }
    *(arr+i+1)=key;
    ne++;
}
void deletebykey(int key)
{
    int pos;
    int flag=0;
    if(ne==0)
    {
        printf("array is empty\n");
        return;
    }
    for(int i=0;i<ne;i++)
    {
        if(*(arr+i)==key){
        flag=1;
        pos=i;
        break;
        }
    }
    if(flag==0)
    printf("element not found\n");
    else{
```

```
        for(int i=pos;i<ne-1;i++)
        *(arr+i)=*(arr+i+1);
        ne--;
    }
}
void searchbypos(int pos)
{
    if(ne==0)
    {
        printf("array is empty\n");
        return;
    }
    if(pos>=1 && pos<=ne)
    printf("element at the position %d is: %d\n",pos,*(arr+pos));
    else
    printf("invalid position\n");
}
void insertbypos(int pos,int key)
{
    int i;
    if(ne==size)
    {
        printf("array is full\n");
        return;
    }
    if(pos>=1 && pos<=ne+1)
    {
        for(i=ne-1;i>=pos;i--){
        *(arr+i+1)=*(arr+i);
        }
        *(arr+i+1)=key;

        ne++;
        return;
    }
    else
    printf("invalid position\n");
}
void reverse()
{
    int j;
    if(ne==0)
    {
        printf("array is empty\n");
        return;
    }
    for(int i=0;i<ne/2;i++)
    {
        j=*(arr+i);
```

```
        *(arr+i)=*(arr+ne-i-1);
        *(arr+ne-i-1)=j;
    }
}
void display()
{
    if(ne==0)
    return;
    for(int i=0;i<ne;i++)
    printf("%d ",*(arr+i));
}
void deletebypos(int pos)
{
    int i;
    if(ne==0)
    {
        printf("array empty!!\n");
        return;
    }
    if(pos>=0 && pos<ne){
    for(i=pos;i<ne;i++)
    *(arr+i)=*(arr+i+1);
    ne--;
    return;
    }
    else
    printf("invalid position\n");
    return;
}
void searchbykey(int key)
{
    int i,flag=0;
    if(ne==0)
    {
        printf("array empty\n");
        return;
    }
    for(i=0;i<ne;i++)
    {
        if(arr[i]==key)
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
    printf("key found at the position :%d\n",i);
    else
    printf("key not found!\n");
```

```
}
void main()
{
    int choice,key,pos;
    printf("enter the array size: ");
    scanf("%d",&size);
    arr=(int*)malloc(size*sizeof(int));
    read();
    printf("main
menu\n1.insertbyorder\n2.deletebykey\n3.searchbypos\n4.insertbypos\n5.rev
erse\n6.deletebyposition\n7.searchbykey\n8.exit\n");
    for(;;)
    {
        printf("\nenter your choice: ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("enter the value: ");
                   scanf("%d",&key);
                   insertbyorder(key);
                   display();
                   break;
            case 2:printf("enter the value: ");
                   scanf("%d",&key);
                   deletebykey(key);
                   display();
                   break;
            case 3:printf("enter the pos: ");
                   scanf("%d",&pos);
                   searchbypos(pos);
                   display();
                   break;
            case 4:printf("enter the pos: ");
                   scanf("%d",&pos);
                   printf("enter the key: ");
                   scanf("%d",&key);
                   insertbypos(pos,key);
                   display();
                   break;
            case 5:reverse();
                   display();
                   break;
            case 6:printf("enter the position: ");
                   scanf("%d",&pos);
                   deletebypos(pos) ;
                   display();
                   break;
            case 7:printf("enter the key to search: ");
```

```
                    scanf("%d",&key);
                    searchbykey(key);
                    display();
                    break;
            case 8:exit(0);
            default:printf("invalid choice\n");
        }

    }
}
```