1. **PROGRAM TO IMPLEMENT DYNAMIC ARRAY AND PERFORM OPERATIONS:**

**TRAVERSING, SEARCHING, INSERTING AND DELETING ON ARRAY.**

//array implementation

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

int main()

{

int \*a;

int cy=1;

int ch,n,i,ele,pos,dl,pd,it,flag,beg,last,mid;

printf("Enter no of elements:");

scanf("%d",&n);

a=(int\*)malloc(n\*sizeof(int));

printf("Enter array:\n");

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

{

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

}

while(cy==1)

{

printf("\n\n\tMENU:\n1. Traversing\n2. Insertion\n3. Deletion\n4. Linear Search\n5. Binary search(for sorted array)\n Enter choice(1-5):");

scanf("%d",&ch);

switch(ch)

{

case 1: printf("Printing entered array alongwith their index number\n");

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

{

printf("%d\t",a[i]);

}

printf("\n");

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

{

printf("(%d)\t",i);

}

break;

case 2: printf("Enter the element to be inserted: ");

scanf("%d",&ele);

printf("\nNow enter the position: ");

scanf("%d",&pos);

if(pos<n)

{

for(i=n;i>=(pos-1);i--)

{

a[i+1]=a[i];

}

a[pos-1]=ele;

}

else

{

a[n]=ele;

}

printf("Element has been successfully entered and now array is:\n");

n=n+1;

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

{

printf("%d\t",a[i]);

}

break;

case 3: printf("Enter the element to be deleted: ");

scanf("%d",&dl);

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

{

if(a[i]==dl)

{

pd=i+1;

}

}

for(i=(pd-1);i<n;i++)

{

a[i]=a[i+1];

}

printf("Element has been successfully deleted and now array is:\n");

n=n-1;

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

{

printf("%d\t",a[i]);

}

break;

case 4 : printf("Enter the element to be searched: ");

scanf("%d",&it);

flag=0;

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

{

if(a[i]==it)

{

flag=1;

printf("Item has been found and its position is %d",(i+1));

}

}

if(flag==0)

{

printf("\nElement not found");

}

break;

case 5: printf(("Enter the element to be searched: ");

scanf("%d",&it);

beg=0;

last=n-1;

mid=(beg+last)/2;

while( beg <= last )

{

if ( a[mid] < it )

{

beg = mid + 1;

}

else if ( a[mid] == it )

{

printf("%d found at location %d.\n", it, mid+1);

break;

}

else

{

last = mid - 1;

mid = (beg + last)/2;

}

}

if ( beg > last )

printf("Not found! %d is not present in the list.\n", it);

break;

}

printf("\nWant to continue(1/0):");

scanf("%d",&cy);

}

getch();

return 0;

}