#include<stdio.h>

#define MAX 1000000

int a[MAX]={0};

//1

void insertAtLast(int a[],int n,int key)

{

a[n]=key;

return;

}

//2

void insertAtFront(int a[],int n,int key)

{

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

a[i]=a[i-1];

a[0]=key;

return;

}

//3

insertAt\_i(int a[],int n,int i,int key)

{

for(int x=n;x>i;x--)

a[x]=a[x-1];

a[i]=key;

return;

}

//4

void deleteFromLast(int a[],int \*n)

{

--\*n;

return;

}

//5

void deleteFromFirst(int a[],int n)

{

for(int x=1;x<n;x++)

a[x-1]=a[x];

return;

}

//6

void deleteFromIndex\_i(int a[],int n,int i)

{

for(int x=i;x<n-1;x++)

{

a[x]=a[x+1];

}

}

//7

int unsortedAscending(int a[],int n)

{

for(int i=0;i<n-1;i++)

{

if(a[i+1]<a[i])

return i+1;

}

return -1;

}

int unsortedDescending(int a[],int n)

{

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

{

if(a[i+1]>a[i])

return i+1;

}

return -1;

}

//8

int sortedAscending(int a[],int n)

{

for(int i=0;i<n-1;i++)

{

if(a[i]<a[i+1])

return i;

}

return -1;

}

int sortedDescending(int a[],int n)

{

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

{

if(a[i]>a[i+1])

return i;

}

return -1;

}

//9

void sortArray(int a[],int n)

{

for(int i=1;i<n;i++)

{

for(int j=0;j<n-i;j++)

{

if(a[j]>a[j+1])

{

int temp=a[j+1];

a[j+1]=a[j];

a[j]=temp;

}

}

}

return;

}

//10

void extractSubset(int a[],int n,int i,int j)

{

for(int x=i;x<=j;x++)

printf("%d ",a[x]);

printf("\n");

return;

}

//11

void deleteSubset(int a[],int n,int i,int j)

{

for(int x=i;x<n;x++)

{

if(x+j>=n)

break;

a[x]=a[x+j];

}

return;

}

//12

void splitArray(int a[],int n)

{

printArray(a,n);

}

//13

void cloneArray(int a[],int n,int b[])

{

for(int i=n;i<2\*n;i++)

a[i]=b[i-n];

return;

}

//14

void shiftLeft(int a[],int n)

{

int curr=a[0];

for(int i=1;i<n;i++)

a[i-1]=a[i];

a[n-1]=curr;

return;

}

//15

void shiftRight(int a[],int n)

{

int curr=a[n-1];

for(int i=n-1;i>0;i--)

{

a[i]=a[i-1];

}

a[0]=curr;

return;

}

//16

void rotateArrayClockwise(int a[],int n,int r)

{

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

shiftRight(a,n);

return;

}

//17

void rotateArrayAntiClockwise(int a[],int n,int r)

{

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

shiftLeft(a,n);

return;

}

//18

int minim(int a[],int n)

{

int ans=0;

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

{

if(a[i]<a[ans])

ans=i;

}

return ans;

}

//19

int maxim(int a[],int n)

{

int ans=0;

for(int i=1;i<n;i++)

{

if(a[i]>a[ans])

ans=i;

}

return ans;

}

//20

void fillPseudoRandom(int a[],int n)

{

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

a[i]=rand();

return;

}

//21

void fillTrueRandom(int a[],int n)

{

srand(time(0));

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

a[i]=rand();

return;

}

//23

void setToZero(int a[],int n)

{

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

a[i]=0;

}

//25

void deletion(int a[],int n)

{

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

free(a[i]);

return;

}

//27

void printArray(int a[],int n)

{

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

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

printf("\n");

}

int main()

{

int n,m;

printf("Enter the initial no of elements in Array\n");

scanf("%d",&n);

printf("Enter the elements\n");

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

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

int key,i,j,r;

int b[100000];

printf("Hello\n");

printf( "Enter choice:\n0. All Operations Completed\n1. Insert item at last\n2. Insert Item At First\n3. Insert Item At Index\n4. DeleteItemFromLast\n5. Delete Item from First\n6. DeleteItemFromIndex\n7. Find Item Unsorted\n8. Find Item Sorted\n9. Sort Array\n10. ExtractSubest\n11. DeleteSubset\n12. Split Into Two Array\n13. m. CloneArray\n14. ShiftLeftArray\n15. ShiftRightArray\n16. RotateArrayClockwise\n17. RotateArrayAntiClockwise\n18. FindMin\n19. FindMax\n20. FillArrayPseudoRandom\n21. FillArrayTrueRandom\n22. IncreaseArraySize\n23. SetArrayToZero\n24. DeleteAllItemOfArray\n25. DeleteArray\n26. AllocateArray\n27. PrintArray\n");

int choice;

do

{

printf("\nEnter choice\n");

scanf("%d",&choice);

switch(choice)

{

case 0:

printf("Operations have been performed successfully\n");

break;

case 1:

printf("Enter the number to be inserted\n");

scanf("%d",&key);

insertAtLast(a,n,key);

n++;

printf("Array has been updated\n");

break;

case 2:

printf("Enter the number to be inserted\n");

scanf("%d",&key);

insertAtFront(a,n,key);

n++;

printf("Array has been updated\n");

break;

case 3:

printf("Enter the number to be inserted and enter the index respectively\n");

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

insertAt\_i(a,n,i,key);

n++;

printf("Array has been updated\n");

break;

case 4:

deleteFromLast(a,&n);

printf("Array has been updated\n");

break;

case 5:

deleteFromFirst(a,n);

n--;

printf("Array has been updated\n");

break;

case 6:

printf("Enter the index to be deleted from\n");

scanf("%d",&i);

deleteFromIndex\_i(a,n,i);

n--;

break;

case 7:

printf("Considering the array to be a ascending array, the Unsorted element is: ");

printf("%d\n",unsortedAscending(a,n));

printf("Considering the array to be a descending array, the Unsorted element is: ");

printf("%d\n",unsortedDescending(a,n));

break;

case 8:

printf("For ascending: ");

printf("%d\n",sortedAscending(a,n));

printf("For descending: ");

printf("%d\n",sortedDescending(a,n));

break;

case 9:

sortArray(a,n);

printf("Array has been Sorted\n");

break;

case 10:

printf("Enter the starting and ending indexes of the subarray in the given array\n");

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

extractSubset(a,n,i,j);

break;

case 11:

printf("Enter the the starting and ending indexed of the subset to be deleted\n");

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

deleteSubset(a,n,i,j);

printf("Required subset has been deleted\n");

n=n-j+i-1;

break;

case 12:

printf("Array after splitting:\n");

n=n/2+1;

splitArray(a,n);

break;

case 13:

printf("Enter %d elements for 2nd array\n",n);

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

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

cloneArray(a,n,b);

n\*=2;

printf("Cloning has been done\n");

break;

case 14:

shiftLeft(a,n);

printf("Array has been shifted left\n");

break;

case 15:

shiftRight(a,n);

printf("Array has been shifted right\n");

break;

case 16:

printf("Enter no of rotations\n");

scanf("%d",&r);

r=r%n;

rotateArrayClockwise(a,n,r);

printf("Array has been rotated\n");

break;

case 17:

printf("Enter no of rotations\n");

scanf("%d",&r);

r=r%n;

rotateArrayAntiClockwise(a,n,r);

printf("Array has been rotated\n");

break;

case 18:

printf("The index containing the minimum element is :%d\n", minim(a,n));

break;

case 19:

printf("The index containing the maximum element is :%d\n", maxim(a,n));

break;

case 20:

fillPseudoRandom(a,n);

printf("Array has been filled with pseudo random numbers\n");

break;

case 21:

fillTrueRandom(a,n);

printf("Array has been filled with true random numbers\n");

break;

case 22:

printf("Enter the new size of Array\n");

scanf("%d",&m);

printf("Size of Array has been changed from %d to %d\n",n,m);

if(m>n)

printf("%d elements are filled with garbage values\n",m-n);

case 23:

setToZero(a,n);

break;

case 24:

n=0;

printf("Elements Of array has been deleted");

break;

case 25:

deletion(a,n);

free(a);

n=0;

break;

case 26:

printf("Enter the new size of array\n");

scanf("%d",&m);

printf("The address of newly allocated array is: %p\n",&b[0]);

break;

case 27:

printArray(a,n);

break;

default:

printf("Error 404\nExit Code status X");

}

}while(choice>=1 && choice<=27);

}