What are the Operations that can be performed on arrays?

Following operations can be performed on arrays:

Traversing

Searching

Insertion

Deletion

Sorting

Merging

1.sTraversing: It is used to access each data item exactly once so that it can be processed.

E.g.

We have linear array A as below:

1 2 3 4 5

10 20 30 40 50

Here we will start from beginning and will go till last element and during this process we will access value of each element exactly once as below:

A [1] = 10

A [2] = 20

A [3] = 30

A [4] = 40

A [5] = 50

2. Searching: It is used to find out the location of the data item if it exists in the given collection of data items.

E.g.

We have linear array A as below:

1 2 3 4 5

15 50 35 20 25

Suppose item to be searched is 20. We will start from beginning and will compare 20 with each element. This process will continue until element is found or array is finished. Here:

1) Compare 20 with 15

20 # 15, go to next element.

2) Compare 20 with 50

20 # 50, go to next element.

3) Compare 20 with 35

20 #35, go to next element.

4) Compare 20 with 20

20 = 20, so 20 is found and its location is 4.

3. Insertion: It is used to add a new data item in the given collection of data items.

E.g.

We have linear array A as below:

1 2 3 4 5

10 20 50 30 15

New element to be inserted is 100 and location for insertion is 3. So shift the elements from 5th location to 3rd location downwards by 1 place. And then insert 100 at 3rd location. It is shown below:

4. Deletion: It is used to delete an existing data item from the given collection of data items.

E.g.

We have linear array A as below:

1 2 3 4 5

10 20 50 40 25 60

The element to be deleted is 50 which is at 3rd location. So shift the elements from 4th to 6th location upwards by 1 place. It is shown below:

After deletion the array will be:

1 2 3 4 5 6

10 20 40 25 60

5. Sorting: It is used to arrange the data items in some order i.e. in ascending or descending order in case of numerical data and in dictionary order in case of alphanumeric data.

E.g.

We have linear array A as below:

1 2 3 4 5

10 50 40 20 30

After arranging the elements in increasing order by using a sorting technique, the array will be:

1 2 3 4 5

10 20 30 40 50

6. Merging: It is used to combine the data items of two sorted files into single file in the sorted form

We have sorted linear array A as below:

1 2 3 4 5 6

10 40 50 80 95 100

And sorted linear array B as below:

1 2 3 4

20 35 45 90

After merging merged array C is as below:

1 2 3 4 5 6 7 8 9 10

10 20 35 40 45 50 80 90 95 100

Menu Driven Program To Perform Various Array Operations  
——Menu——-   
1.Create   
2.Display  
3.Insert  
4.Delete   
5.Search   
6.Merge   
7.Sort   
8.Exit

#include<stdlib.h>

int a[20],b[20],c[40];

int m,n,p,val,i,j,key,pos,temp;

/\*Function Prototype\*/

void create();

void display();

void insert();

void del();

void search();

void merge();

void sort();

int main()

{

int choice;

do{

printf("\n\n--------Menu-----------\n");

printf("1.Create\n");

printf("2.Display\n");

printf("3.Insert\n");

printf("4.Delete\n");

printf("5.Search\n");

printf("6.Sort\n");

printf("7.Merge\n");

printf("8.Exit\n");

printf("-----------------------");

printf("\nEnter your choice:\t");

scanf("%d",&choice);

switch(choice)

{

case 1: create();

break;

case 2:

display();

break;

case 3:

insert();

break;

case 4:

del();

break;

case 5:

search();

break;

case 6:

sort();

break;

case 7:

merge();

break;

case 8:

exit(0);

break;

default:

printf("\nInvalid choice:\n");

break;

}

}while(choice!=8);

return 0;

}

void create() //creating an array

{

printf("\nEnter the size of the array elements:\t");

scanf("%d",&n);

printf("\nEnter the elements for the array:\n");

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

{

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

}

}//end of create()

void display() //displaying an array elements

{

int i;

printf("\nThe array elements are:\n");

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

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

}

}//end of display()

void insert() //inserting an element in to an array

{

printf("\nEnter the position for the new element:\t");

scanf("%d",&pos);

printf("\nEnter the element to be inserted :\t");

scanf("%d",&val);

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

{

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

}

a[pos]=val;

n=n+1;

}//end of insert()

void del() //deleting an array element

{

printf("\nEnter the position of the element to be deleted:\t");

scanf("%d",&pos);

val=a[pos];

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

{

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

}

n=n-1;

printf("\nThe deleted element is =%d",val);

}//end of delete()

void search() //searching an array element

{

printf("\nEnter the element to be searched:\t");

scanf("%d",&key);

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

{

if(a[i]==key)

{

printf("\nThe element is present at position %d",i);

break;

}

}

if(i==n)

{

printf("\nThe search is unsuccessful");

}

}//end of serach()

void sort() //sorting the array elements

{

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

{

for(j=0;j<n-1-i;j++) { if(a[j]>a[j+1])

{

temp=a[j];

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

a[j+1]=temp;

}

}

}

printf("\nAfter sorting the array elements are:\n");

display();

}//end of sort

void merge() //merging two arrays

{

printf("\nEnter the size of the second array:\t");

scanf("%d",&m);

printf("\nEnter the elements for the second array:\n");

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

{

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

}

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

{

c[j]=a[i];

}

for(i=0;i<m;i++,j++)

{

c[j]=b[i];

}

p=n+m;

printf("\nArray elements after merging:\n");

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

{

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

}

}//end of merge()