**Functions and Pointers**

**Session VIII:**

**Searching an Array:** Write a C program to search for an element ‘a’ in the array. (Linear Search)

#include<stdio.h>

int main()

{

int n,search\_element,i;

//printf("Enter size\n");

scanf("%d",&n);

int a[n];

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

{

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

}

scanf("%d",&search\_element);

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

{

if(a[i]==search\_element)

{

printf("%d is present in the array",search\_element);

break;

}

}

if(i==n)

printf("%d is not present in the array",search\_element);

return 0;

}

**Implementation of Binary Search:** Write a C program to implement Binary Search Algorithm.

#include<stdio.h>

#include<stdlib.h>

int BinarySearch(int first, int last,int \*a,int x);

int main()

{

int n, \*array,search,c;

int lower,upper;

printf("Enter the number of elements :\n");

scanf("%d",&n);

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

printf("Enter the elements :\n");

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

{

scanf("%d",(array+c));

}

printf("Enter the element to be searched :\n");

scanf("%d",&search);

lower=0;

upper=n-1;

BinarySearch(lower,upper,array,search);

return 0;

}

int BinarySearch(int first,int last,int \*a,int x)

{

int mid;

// first=0,last=n-1;

while(first<=last)

{

mid=(first+last)/2;

if(a[mid]<x)

first=mid+1;

else if(a[mid]==x){

printf("The element %d is in position %d",x,mid);

break;

}

else

last=mid-1;

mid=(first+last)/2;

}

if(first>last)

{

printf("The element %d is nor present in the array",x);

}

return 0;

}

**Sorting Algorithms session IX  
SELECTION SORT**  
  
Write a C program to perform selection sort on an array of n elements.

#include<stdio.h>

int main(){

int s,i,j,temp,a[20],k;

printf("Enter the number of elements in the array\n");

scanf("%d",&s);

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

{

printf("Enter element %d\n",i+1);

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

}

printf("Selection sort.\n");

printf("array before sorting:\n");

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

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

printf("\n");

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

for(j=i+1;j<s;j++){

if(a[i]>a[j]){

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

printf("After Iteration %d\n",i+1);

for(k=0;k<s;k++)

{

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

}

printf("\n");

}

printf("array after sorting:\n");

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

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

return 0;

}

**INSERTION SORT**  
  
Write a C program to perform insertion sort on an array of n elements.

#include<stdio.h>

int main(){

int i,j,s,k,temp,a[20];

printf("Enter the number of elements in the array\n");

scanf("%d",&s);

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

{

printf("Enter element %d\n",i+1);

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

}

printf("Insertion sort.\n");

printf("array before sorting:\n");

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

{

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

}

printf("\n");

for(i=1;i<s;i++){

temp=a[i];

j=i-1;

while((temp<a[j])&&(j>=0)){

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

j=j-1;

}

a[j+1]=temp;

printf("After Iteration %d\n",i);

for(k=0;k<s;k++)

{

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

}

printf("\n");

}

printf("array after sorting:\n");

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

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

return 0;

}

Bubble Sort:

#include<stdio.h>

int main()

{

int n,arr[20],i,j,temp,xchanges,c;

printf("Enter the number of elements :\n");

scanf("%d",&n);

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

{

printf("Enter element %d\n",i+1);

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

}

printf("Unsorted list is :\n");

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

{

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

}

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

{

xchanges=0;

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

{

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

{

temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

xchanges++;

}

}

printf("\nAfter Pass %d elements are :",i+1);

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

{

printf("%d ",arr[c]);

}

if(xchanges==0)

break;

}

printf("\nSorted list is :\n");

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

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

return 0;

}

Check the array in ascending order:

Write a program to find whether the given array is sorted in ascending order.

#include<stdio.h>

int main()

{

int n,i,j,istrue;

scanf("%d",&n);

int a[n];

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

{

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

}

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

{

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

{

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

{

istrue=1;

continue;

}

else

{

istrue=0;

break;

}

}

if(istrue)

continue;

else

break;

}

if(istrue)

printf("yes");

else

printf("no");

return 0;

}

Write a program to find whether the given array is sorted in descending order.

**#include<stdio.h>**

**int main()**

**{**

**int n,i,j,istrue;**

**scanf("%d",&n);**

**int a[n];**

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

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

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

**{**

**for(j=i+1;j<n;j++)**

**{**

**if(a[i]>a[j])**

**{**

**istrue=1;**

**continue;**

**}**

**else**

**{**

**istrue=0;**

**break;**

**}**

**}**

**if(istrue)**

**continue;**

**else**

**break;**

**}**

**if(istrue)**

**printf("yes");**

**else**

**printf("no");**

**return 0;**

**}**

[check whether the given array is sorted in ascending or descending order](http://stackoverflow.com/questions/29605821/check-whether-the-given-array-is-sorted-in-ascending-or-descending-order)

#include<stdio.h>

int main()

{

int arr[15],n,i,j,flag1=1,flag2=1;

scanf("%d",&n);

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

{

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

}

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

{

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

{

if(arr[i]<arr[j])

flag1=0;

break;

}

}

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

{

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

{

if(arr[i]>arr[j])

flag2=0;

break;

}

}

if(flag1 || flag2)

printf("yes");

else

printf("no");

return 0;

}