1. Write a program to read and print elements of an array.

IPO

| **Component** | **Description** |
| --- | --- |
| **Input** | A string (word) entered by the user via scanf() |
| **Process** | The program stores the string in a character array |
| **Output** | Prints the same string using printf() |

CODE

#include<stdio.h>

void main()

{

char str[10];

int i;

scanf("%s",str);

printf("PRINT WORD=%s",str);

}

OUTPUT



1. Write a program to find the sum of elements of an array.

IPO

| **Component** | **Description** |
| --- | --- |
| **Input** | Two integers entered by the user using scanf() |
| **Process** | Store the integers in an array and calculate their sum |
| **Output** | Print the total sum using printf() |

CODE

#include<stdio.h>

void main()

{

int a[2],i,sum=0;

for(i=0;i<2;i++)

{

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

}

for(i=0;i<2;i++)

{

sum=sum+a[i];

}

printf("sum=%d",sum);

}

OUTPUT



1. Write a program to find the maximum and minimum element in an array.

IPO

| **Component** | **Description** |
| --- | --- |
| **Input** | 5 integers entered by the user and stored in an array |
| **Process** | Compare elements to find the **maximum** and **minimum** values |
| **Output** | Print the maximum and minimum values using printf() |

CODE

#include<stdio.h>

void main()

{

int a[5],i,max,min;

for(i=0;i<5;i++)

{

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

}

max=a[0];

min=a[0];

for(i=0;i<5;i++)

{

if(max<a[i])

{

max=a[i];

}

}

for(i=0;i<5;i++)

{

if(min>a[i])

{

min=a[i];

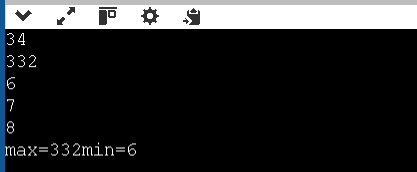
}

}

printf("max=%dmin=%d",max,min);

}

OUTPUT



1. Write a program to reverse an array.

IPO

| **Part** | **Description** |
| --- | --- |
| **Input** | A predefined string "welcome" stored in str |
| **Process** | Count number of characters and reverse the string into str2 |
| **Output** | Print the string length and the reversed string |

CODE

#include<stdio.h>

void main()

{

char str[7]="welcome";

char str2[7];

int i,count=0;

for(i=0;i<7;i++)

{

if(str[i]!='\0')

count++;

}

printf("%d\n",count);

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

{

str2[i]=str[count-1-i];

}

printf("%s\n",str2);

}

OUTPUT



1. Write a program to search for an element in an array (linear search).

IPO

| **Part** | **Description** |
| --- | --- |
| **Input** | Number of elements, array elements, key to find |
| **Process** | Compare each element of array with key |
| **Output** | Print position if found, or not found message |

CODE

#include <stdio.h>

void main()

{

int a[100], n, i, key, found = 0;

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

scanf("%d", &n);

printf("Enter %d elements:\n", n);

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

{

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

}

printf("Enter the element to search: ");

scanf("%d", &key);

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

{

if(a[i] == key)

{

found = 1;

printf("Element %d found at position %d\n", key, i + 1); // 1-based position

break;

}

}

if(!found)

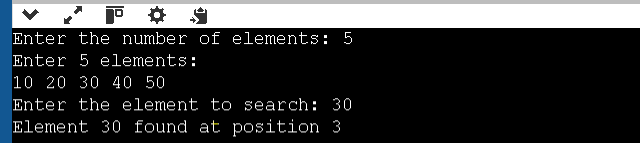
{

printf("Element %d not found in the array.\n", key);

}

}

OUPUT



1. Write a program to sort an array in ascending order.

| **Part** | **Description** |
| --- | --- |
| **Input** | Number of elements, and the array itself |
| **Process** | Compare and swap elements to sort the array |
| **Output** | Sorted array in ascending order |

CODE

#include <stdio.h>

void main()

{

int a[100], n, i, j, temp;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

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

{

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

}

// Bubble Sort

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

{

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

{

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

{

// Swap a[j] and a[j+1]

temp = a[j];

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

a[j + 1] = temp;

}

}

}

printf("Array in ascending order:\n");

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

{

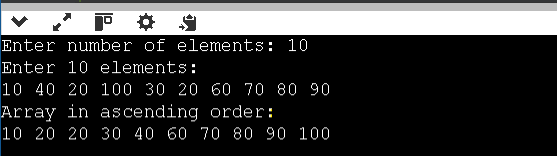
printf("%d ", a[i]);

}

printf("\n");

}

OUTPUT



1. Write a program to insert an element in an array.

IPO

| **Part** | **Description** |
| --- | --- |
| **Input** | Size of array, array elements, position to insert, and the value |
| **Process** | Shift elements to the right, insert new value at correct position |
| **Output** | Print array after insertion |

CODE

#include <stdio.h>

void main()

{

int a[100], n, i, pos, value;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

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

{

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

}

printf("Enter the position to insert (1 to %d): ", n+1);

scanf("%d", &pos);

printf("Enter the value to insert: ");

scanf("%d", &value);

// Shift elements to the right

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

{

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

}

// Insert the value

a[pos - 1] = value;

n++; // Increase array size

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

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

{

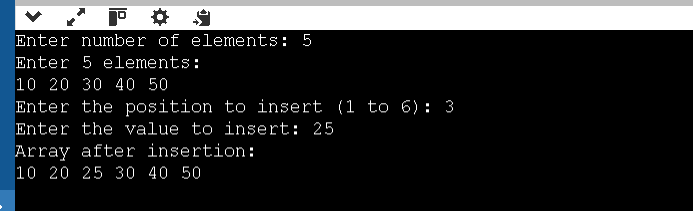
printf("%d ", a[i]);

}

printf("\n");

}

OUTPUT



1. Write a program to delete an element from an array.

IPO

| **Part** | **Description** |
| --- | --- |
| **Input** | Number of elements, the array, and the position to delete |
| **Process** | Shift elements left from the deletion point and reduce size |
| **Output** | Array after deleting the specified element |

CODE

#include <stdio.h>

void main()

{

int a[100], n, i, pos;

printf("Enter number of elements: ");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

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

{

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

}

printf("Enter the position to delete (1 to %d): ", n);

scanf("%d", &pos);

if(pos < 1 || pos > n)

{

printf("Invalid position!\n");

}

else

{

// Shift elements to the left

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

{

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

}

n--; // Reduce array size

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

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

{

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

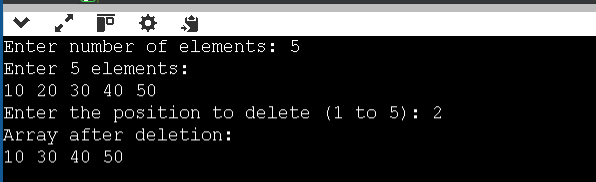
}

printf("\n");

}

}

OUTPUT



1. Write a program to find the frequency of elements in an array.

IPO

| **Part** | **Description** |
| --- | --- |
| **Input** | Number of elements and the array elements |
| **Process** | Count how many times each unique element appears in the array |
| **Output** | Display each element and how many times it appears |

CODE

#include <stdio.h>

void main()

{

int a[10],n,i,j,count;

printf("Enter number of elements: ");

scanf("%d",&n);

printf("Enter %d numbers:\n", n);

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

{

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

}

printf("Frequencies of elements:\n");

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

{

count = 1;

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

{

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

{

count=0;

break;

}

}

if(count==0)

continue;

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

{

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

{

count++;

}

}

printf("%d occurs %d times\n",a[i],count);

}

}

1. Write a program to merge two arrays.

IPO

| **Part** | **Description** |
| --- | --- |
| **Input** | Two arrays entered by the user |
| **Process** | Copy all elements from both arrays into a third array |
| **Output** | Display the merged array |

CODE

#include <stdio.h>

void main()

{

int a[50], b[50], merge[100];

int n1, n2, i, k;

// Input first array

printf("Enter number of elements in first array: ");

scanf("%d", &n1);

printf("Enter %d elements:\n", n1);

for(i = 0; i < n1; i++)

{

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

}

// Input second array

printf("Enter number of elements in second array: ");

scanf("%d", &n2);

printf("Enter %d elements:\n", n2);

for(i = 0; i < n2; i++)

{

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

}

// Merge both arrays into 'merge' array

for(i = 0; i < n1; i++)

{

merge[i] = a[i];

}

for(k = 0; k < n2; k++)

{

merge[i] = b[k];

i++;

}

// Print merged array

printf("Merged array:\n");

for(i = 0; i < n1 + n2; i++)

{

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

}

printf("\n");

}

OUTPUT

