

Basic Programs on simple arrays and 1 Dimensional Arrays:

1D
Arrays

Program to input 10 numbers in an array and display only the even numbers if present in the array.

Program to input 5 numbers in an array and print all the numbers from the backside of the array. Example: 12 18 16 Output: 16 18 12

Program finds the highest and lowest elements in an array.

Program finds the highest and lowest elements in an array.

Example 1

Program to input 10 numbers in an array and display only the even numbers if present in the array.

```
#include <stdio.h>
#include <conio.h>

int main()
{
    int a[10], i;

    printf("Enter 10 numbers\n");
    for(i=0; i<10; i++)
    {
        scanf("%d",&a[i]);
    }

    printf("List of even numbers\n");
    for(i=0; i<10; i++)
    {
        if(a[i]%2==0) ←
        {
            printf("%d ",a[i]);
        }
    }
    return 0;
}
```

Output

```
Enter 10 numbers
11
15
28
31
49
54
72
81
93
14
List of even numbers
28 54 72 14
```

Example 2

Program to input 5 numbers in an array and print all the numbers from the backside of the array. Example: 12 18 16 Output: 16 18 12

```
#include <stdio.h>
#include <conio.h>

int main()
{
    int a[5], i;

    printf("Enter 5 numbers\n");
    for(i=0; i<5; i++)
    {
        scanf("%d",&a[i]);
    }

    ←
    for(i=4; i>=0; i--)
    {
        printf("%d ",a[i]);
    }
    return 0;
}
```

Output

```
Enter 5 numbers
48
21
```

97
64
53
53 64 97 21 48

Example 3: program finds the highest and lowest elements in an array.

```
1 #include<stdio.h>
2 #define SIZE 10
3
4 int main()
5 {
6     int my_arr[SIZE] = {34,56,78,15,43,71,89,34,70,91};
7     int i, max, min;
8     max = min = my_arr[0]; = 34
9
10    for(i = 0; i < SIZE; i++)
11    {
12        // if value of current element is greater than previous value
13        // then assign new value to max
14        if(my_arr[i] > max)
15        {
16            max = my_arr[i];
17        }
18
19        // if the value of current element is less than previous element
20        // then assign new value to min
21        if(my_arr[i] < min)
22        {
23            min = my_arr[i];
24        }
25    }
26
27    printf("Lowest value = %d\n", min);
28    printf("Highest value = %d", max);
29
30    // signal to operating system everything works fine
31    return 0;
32 }
33 }
```

Handwritten annotations:

- Blue arrows pointing to the array initialization and the for loop.
- Handwritten "Min" and "Max" labels with values: Min 15, Max 91 (with 78 crossed out).
- Handwritten comparisons: $34 > 34$ and $56 > 34$.
- Handwritten "X" next to the min update logic, with a note: $34 < 34$.
- Handwritten "Min" and "Max" labels with values: Min 34, Max 56.

Output:

```
1|Lowest value = 15
2|Highest value = 91
```

Example 4: program finds the highest and lowest elements in an array.

Passing 1-D array elements to a function

We can pass elements of 1-D array just like any normal variables. The following example demonstrates the same.

```
1 #include<stdio.h>
2 void odd_or_even(int a);
3
4 int main()
5 {
6     int my_arr[] = {13,56,71,38,93}, i;
7
8     for(i = 0; i < 5; i++)
9     {
10        // passing one element at a time to odd_or_even() function
11        odd_or_even(my_arr[i]);
12    }
13
14    // signal to operating system program ran fine
15    return 0;
16 }
17
18 void odd_or_even(int a)
19 {
20     if(a % 2 == 0)
21     {
22         printf("%d is even\n", a);
23     }
24     else
25     {
26         printf("%d is odd\n", a);
27     }
28 }
29 }
```

Handwritten annotations:

- Blue arrow pointing to `my_arr[]` in line 6.
- Blue arrow pointing to `my_arr[i]` in line 11.
- Handwritten `OE (13)` in blue below line 11.
- Handwritten `13` in blue below line 16.
- Blue arrow pointing to the `printf` statement in line 26.

Expected Output:

```
1 13 is odd
2 56 is even
3 71 is odd
4 38 is even
5 93 is odd
```