

1. Integer Array

Aim:

To write a program that reads the elements of an array input by the user and prints the array.

Algorithm:

Step 1: Start

Step 2: Read an array variable with size greater than or equal to 10.

Step 3: Read the size of the array as input from the user.

Step 4: Run a for loop from index=0 to index=size-1, incrementing the index by 1 after each iteration. Read an element into the array for each iteration.

Step 5: Run the for loop again in the same fashion reinitializing the loop variable from 0. Print the corresponding element for each iteration.

Step 6: End

Code:

```
//to read and print elements of an array
#include <stdio.h>

int main() {
    int arr[20], size;
    printf("Input size: ");
    scanf("%d",&size);
    printf("Enter numbers: ");
    for (int i=0; i<size; i++){
        scanf("%d",&arr[i]);
    }
    //for printing
    printf("Array: ");
    for (int i=0; i<size; i++)
        printf(" %d",arr[i]);

    return 0;
}
```

Output:

```
> gcc -o gal.out gal.c
> ./gal.out
Input size: 10
Enter numbers: 1
2
3
4
5
6
7
8
9
10
Array: 1 2 3 4 5 6 7 8 9 10
```

Result:

A program for reading and printing an integer array is written and executed.

2. Negative Elements of an Array

Aim:

To write a program that reads an array from the user and prints only the negative elements of the array.

Algorithm:

Step 1: Start

Step 2: Read an array variable with size greater than or equal to 10.

Step 3: Read the size of the array as input from the user.

Step 4: Run a for loop from index=0 to index=size-1, incrementing the index by 1 after each iteration. Read an element into the array for each iteration.

Step 5: Run the for loop again in the same fashion reinitializing the loop variable from 0.

Step 6: Check if the corresponding element for each iteration is less than 0. If yes, print the corresponding element for each iteration.

Step 7: End

Code:

```
//to read and print only the negative elements of an array
#include <stdio.h>

int main() {
    int arr[10], size;
    printf("Input size: ");
    scanf("%d",&size);
    printf("Enter numbers: ");
    for (int i=0; i<size; i++){
        scanf("%d",&arr[i]);
    }
    //for printing
    printf("Array: ");
    for (int i=0; i<size; i++){
        if (arr[i]<0)
            printf(" %d",arr[i]);
    }
    printf("\n");

    return 0;
}
```

Output:

```
> gcc -o qa2.out qa2.c
> ./qa2.out
Input size: 10
Enter numbers: -1
-10
100
5
61
-2
-23
8
-90
51
Array: -1 -10 -2 -23 -90
```

Result:

A program for reading an array and printing its negative elements is written and executed.

3. Maximum and Minimum of an Array

Aim:

To write a program that reads an array and prints the maximum and minimum elements of the array.

Algorithm:

Step 1: Start

Step 2: Read an array variable with size greater than or equal to 10.

Step 3: Read the size of the array as input from the user.

Step 4: Run a for loop from index=0 to index=size-1, incrementing the index by 1 after each iteration. Read an element into the array for each iteration.

Step 5: Read two variables (maximum and minimum) and set their values to be the first element of the given array.

Step 6: Run the for loop again in the same fashion reinitializing the loop variable to 0.

Step 7: If in any iteration, the corresponding element is greater than the maximum or lesser than the minimum, change the value of the maximum (or minimum) to the element.

Step 8: Print the maximum and minimum.

Step 9: End

Code:

```
//to read and print the maximum and minimum elements of an array
#include <stdio.h>

int main() {
    int arr[10], size;
    printf("Input size: ");
    scanf("%d",&size);
    printf("Enter numbers: ");
    for (int i=0; i<size; i++){
        scanf("%d",&arr[i]);
    }

    //for finding the maximum and minimum
    int min=arr[0], max=arr[0];
    for (int i=0; i<size; i++){
        if (arr[i]>max)
            max=arr[i];
        if (arr[i]<min)
            min=arr[i];
    }

    //for printing the max and min
    printf("Maximum: %d\nMinimum: %d\n",max,min);

    return 0;
}
```

Output:

```
> gcc -o qa3.out qa3.c
> ./qa3.out
Input size: 5
Enter numbers: 3
5
1
7
9
Maximum: 9
Minimum: 1
```

Result:

A program for reading an array and printing its maximum and minimum elements is written and executed.

4. Odd and Even elements in an Array

Aim:

To write a program that reads an array and separately counts the number of odd and even elements of the array.

Algorithm:

Step 1: Start

Step 2: Read an array variable with size greater than or equal to 10.

Step 3: Read the size of the array as input from the user.

Step 4: Run a for loop from index=0 to index=size-1, incrementing the index by 1 after each iteration. Read an element into the array for each iteration.

Step 5: Read two variables (count_odd and count_even) and set their values to be 0.

Step 6: Run the for loop again in the same fashion reinitializing the loop variable to 0.

Step 7: If in any iteration, the corresponding element is divisible by 2, increment the count of even elements by 1.

If the element is not divisible by 2, increment the count of odd elements by 1.

Step 8: Print the individual counts of the odd and even elements separately.

Step 9: End

Code:

```
//to count the number of odd and even elements of an array
```

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

```
int main() {
    int arr[10], size;
    printf("Input size: ");
    scanf("%d",&size);
    printf("Enter numbers: ");
    for (int i=0; i<size; i++){
        scanf("%d",&arr[i]);
    }
    //for counting the odd and even elements
    int ct_odd=0, ct_even=0;
    for (int i=0; i<size; i++){
        if (arr[i]%2==0)
            ct_even++;
        else
            ct_odd++;
    }
    //for printing the max and min
    printf("Even: %d\nOdd: %d\n",ct_even,ct_odd);

    return 0;
}
```

Output:

```
> gcc -o qa4.out qa4.c
> ./qa4.out
Input size: 9
Enter numbers: 1
2
3
4
5
6
7
8
9
Even: 4
Odd: 5
```

Result:

A program to read an array and count the number of even and odd elements in it, is written and executed.

5. Copying an Array

Aim:

To write a program to read an array and copy all the elements in that array to a second array.

Algorithm:

Step 1: Start

Step 2: Read an array variable with size greater than or equal to 10.

Step 3: Read the size of the array as input from the user.

Step 4: Run a for loop from index=0 to index=size-1, incrementing the index by 1 after each iteration. Read an element into the array for each iteration.

Step 5: Read a new array variable with the same size as the first array.

Step 6: Run the for loop again in the same fashion, reinitializing the loop variable to 0, and for each iteration, assign the value of the element in the first array to the corresponding position in the new array.

Step 7: Print both the arrays one by one using for loops in the same format that has been used above.

Step 8: End

Code:

```
//to copy all elements of an array into a second array
```

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

```
int main() {  
    int arr[10], size;  
    printf("Input size: ");  
    scanf("%d",&size);  
    printf("Enter numbers: ");  
    for (int i=0; i<size; i++){  
        scanf("%d",&arr[i]);  
    }
```

```
//for copying into a new array
```

```
int new[10];  
for (int i=0; i<size; i++){  
    new[i]=arr[i];  
}
```

```
//for printing
```

```
printf("Array1: ");  
for (int i=0; i<size; i++)  
    printf(" %d",arr[i]);  
printf("\n");
```

```
//for printing
```

```
printf("Array2: ");  
for (int i=0; i<size; i++)
```

```
    printf(" %d", new[i]);  
    printf("\n");  
  
    return 0;  
}
```

Output:

```
> gcc -o qa5.out qa5.c  
> ./qa5.out  
Input size: 10  
Enter numbers: 1  
3  
8  
-4  
98  
-23  
12  
108  
9  
86  
Array1:  1 3 8 -4 98 -23 12 108 9 86  
Array2:  1 3 8 -4 98 -23 12 108 9 86
```

Result:

A program for copying all the elements of an array into a second array is written and executed.