

## PRACTICE LAB ASSIGNMENT 8

1. Twenty-Five numbers are entered from the keyboard into an array. Write a program to find out how many of them are positive, how many are negative, how many are even and how many are odd.

### CODE

```
#include <stdio.h>

#define SIZE 25

int main (void)
{
    int n[SIZE], i, P = 0, N = 0, E = 0, O = 0;

    printf("\nEnter 25 numbers to find out how many are POSITIVE, NEGATIVE, EVEN, ODD");

    for(i = 0; i < SIZE; ++i)
    {
        printf("\nEnter the value for the array: ");

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

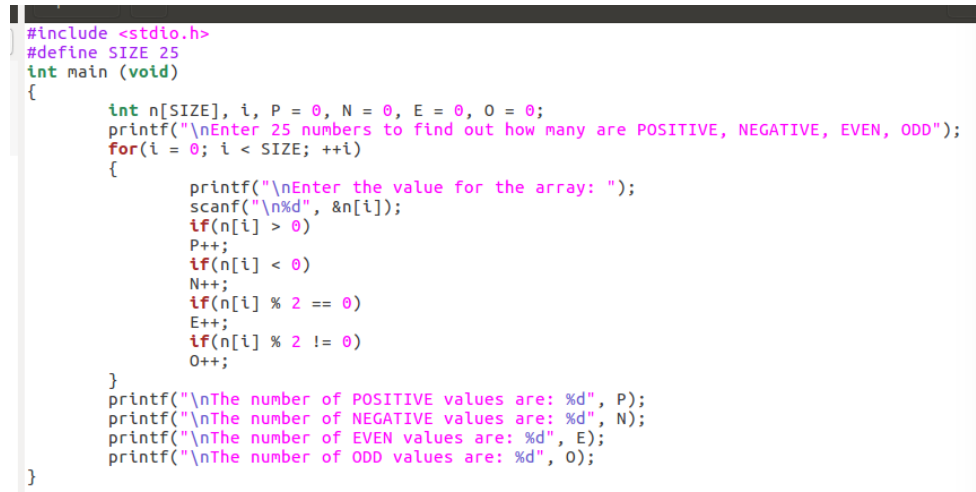
        if(n[i] > 0)
            P++;

        if(n[i] < 0)
            N++;

        if(n[i] % 2 == 0)
            E++;

        if(n[i] % 2 != 0)
            O++;
    }

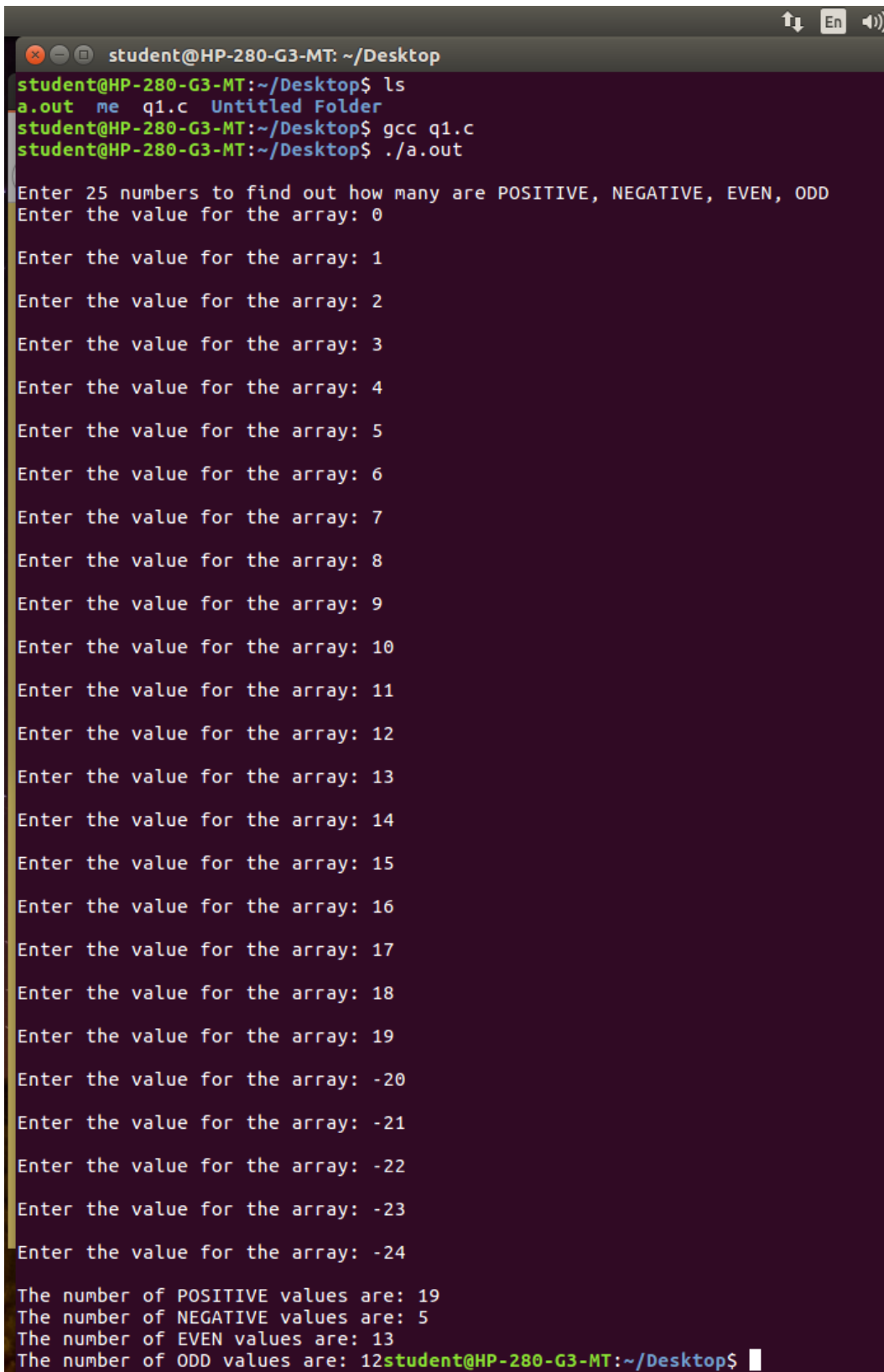
    printf("\nThe number of POSITIVE values are: %d", P);
    printf("\nThe number of NEGATIVE values are: %d", N);
    printf("\nThe number of EVEN values are: %d", E);
    printf("\nThe number of ODD values are: %d", O);
}
```

A screenshot of a code editor showing the same C program as the one in the previous block. The code is color-coded: keywords like 'include', 'define', 'main', 'void', 'int', 'for', 'if', 'scanf', 'printf', and 'return' are in blue; variables and constants are in black; and string literals are in red. The program logic is identical to the one provided in the text blocks.

```
#include <stdio.h>
#define SIZE 25
int main (void)
{
    int n[SIZE], i, P = 0, N = 0, E = 0, O = 0;
    printf("\nEnter 25 numbers to find out how many are POSITIVE, NEGATIVE, EVEN, ODD");
    for(i = 0; i < SIZE; ++i)
    {
        printf("\nEnter the value for the array: ");
        scanf("\n%d", &n[i]);
        if(n[i] > 0)
            P++;
        if(n[i] < 0)
            N++;
        if(n[i] % 2 == 0)
            E++;
        if(n[i] % 2 != 0)
            O++;
    }
    printf("\nThe number of POSITIVE values are: %d", P);
    printf("\nThe number of NEGATIVE values are: %d", N);
    printf("\nThe number of EVEN values are: %d", E);
    printf("\nThe number of ODD values are: %d", O);
}
```

}

## SS of the OUTPUT



```
student@HP-280-G3-MT: ~/Desktop
student@HP-280-G3-MT:~/Desktop$ ls
a.out  me  q1.c  Untitled Folder
student@HP-280-G3-MT:~/Desktop$ gcc q1.c
student@HP-280-G3-MT:~/Desktop$ ./a.out

Enter 25 numbers to find out how many are POSITIVE, NEGATIVE, EVEN, ODD
Enter the value for the array: 0

Enter the value for the array: 1

Enter the value for the array: 2

Enter the value for the array: 3

Enter the value for the array: 4

Enter the value for the array: 5

Enter the value for the array: 6

Enter the value for the array: 7

Enter the value for the array: 8

Enter the value for the array: 9

Enter the value for the array: 10

Enter the value for the array: 11

Enter the value for the array: 12

Enter the value for the array: 13

Enter the value for the array: 14

Enter the value for the array: 15

Enter the value for the array: 16

Enter the value for the array: 17

Enter the value for the array: 18

Enter the value for the array: 19

Enter the value for the array: -20

Enter the value for the array: -21

Enter the value for the array: -22

Enter the value for the array: -23

Enter the value for the array: -24

The number of POSITIVE values are: 19
The number of NEGATIVE values are: 5
The number of EVEN values are: 13
The number of ODD values are: 12student@HP-280-G3-MT:~/Desktop$
```

2.Implement the following procedure to generate prime numbers from 1 to 100 into a program. This procedure is called sieve of Eratosthenes.

Step 1Fill an array num[100]with numbers from 1 to 100.

Step 2 Starting with the second entry in the array, set all its multiples to zero.

Step 3 Proceed to the next non-zero element and set all its multiples to zero.

Step 4 Repeat step 3 till you have set up the multiples of all the non-zero elements to zero.

Step 5 At the conclusion of step 4, all the non-zero entries left in the array would be prime numbers, so print out these numbers.

#### CODE

```
#include <stdio.h>

int perfect(int);

int main()
{
    int n;
    printf("\nThe list of perfect numbers between 1 and 1000 are:");
    perfect(n);
    return 0;
}

int perfect(int n)
{
    int i, sum;
    for(n = 1; n <= 1000; n++)
    {
        sum = 0;
        for(i = 1; i < n; i++)
        {
            if(n%i == 0)
            {
                sum += i;
            }
        }
    }
}
```

```

if(sum == n)
{
    printf("\n%d is a PERFECT number", n);
    printf("\nIt's factors are: ");
    {
        for(i = 1; i < n; i++)
        {
            if(n%i == 0)
            {
                printf("%d ", i);
            }
        }
        printf("\n");
    }
}
}
}

```

**SS of the OUTPUT**

**3. Write a program to copy the contents of one array into another in the reverse order.**

**CODE**

```

#include <stdio.h>

int main (void)

```

```

{
int a;

printf("\nEnter the size of your array: ");
scanf("%d", &a);

int n[a], i, m[a], j;

for(i = 0; i < a; ++i)
{
printf("\nEnter the value for the array: ");
scanf("\n%d", &n[i]);
}

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

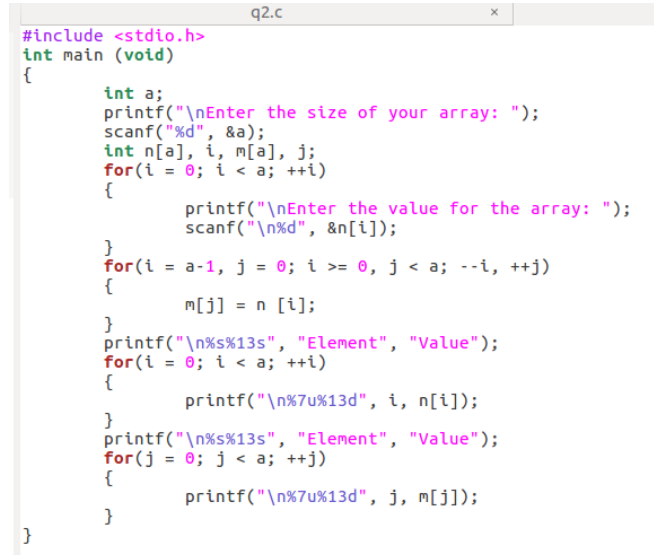
printf("\n%s%13s", "Element", "Value");

for(i = 0; i < a; ++i)
{
printf("\n%7u%13d", i, n[i]);
}

printf("\n%s%13s", "Element", "Value");

for(j = 0; j < a; ++j)
{
printf("\n%7u%13d", j, m[j]);
}
}

```



```

q2.c
#include <stdio.h>
int main (void)
{
    int a;
    printf("\nEnter the size of your array: ");
    scanf("%d", &a);
    int n[a], i, m[a], j;
    for(i = 0; i < a; ++i)
    {
        printf("\nEnter the value for the array: ");
        scanf("\n%d", &n[i]);
    }
    for(i = a-1, j = 0; i >= 0, j < a; --i, ++j)
    {
        m[j] = n [i];
    }
    printf("\n%s%13s", "Element", "Value");
    for(i = 0; i < a; ++i)
    {
        printf("\n%7u%13d", i, n[i]);
    }
    printf("\n%s%13s", "Element", "Value");
    for(j = 0; j < a; ++j)
    {
        printf("\n%7u%13d", j, m[j]);
    }
}

```

SS of the OUTPUT

```
student@HP-280-G3-MT: ~/Desktop
student@HP-280-G3-MT:~$ pwd
/home/student
student@HP-280-G3-MT:~$ cd Desktop
student@HP-280-G3-MT:~/Desktop$ ls
a.out  me  Q1  q1.c  q2.c  q3.c  Untitled Folder
student@HP-280-G3-MT:~/Desktop$ gcc q3.c
student@HP-280-G3-MT:~/Desktop$ ./a.out

Enter the size of your array: 5

Enter the value for the array: 1

Enter the value for the array: 2

Enter the value for the array: 3

Enter the value for the array: 4

Enter the value for the array: 5

Element      Value
0            1
1            2
2            3
3            4
4            5
Element      Value
0            5
1            4
2            3
3            2
4            1student@HP-280-G3-MT:~/Desktop$
```

4. Write a program to insert a number in an array that is already sorted in ascending order. You need to take the size of the array one extra than the original size.

CODE

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

```

int fibonacci(int, int, int, int);

int main()
{
    int t, a = 0, b = 1, n = a + b;

    printf("\nEnter the number of terms of the Fibonacci series you need: ");

    scanf("%d", &t);

    printf("A Fibonacci series of %d terms will be: 0, 1, ", t);

    fibonacci(t, a, b, n);

    return 0;
}

int fibonacci(int t, int a, int b, int n)
{
    int i;

    for(i = 1; i <= (t - 2); i++)
    {
        printf("%d, ", n);

        a = b;

        b = n;

        n = a + b;
    }
}

```

## SS of the OUTPUT

5. Read in 20 numbers each of which is between 10 and 100, inclusive. As each number is read, store it only if it is not a duplicate of a number already read. Use the smallest possible array to solve this problem.

## CODE

```

#include<stdio.h>

int multiple(int, int);

```

```

int main()
{
    int n, m;
    printf("\nEnter any integer: ");
    scanf("%d", &n);
    printf("\nEnter an integer to check if it's a multiple of the previous number: ");
    scanf("%d", &m);
    printf("%d", multiple(n, m));
    return 0;
}

int multiple(int n, int m)
{
    if(m%n == 0)
        return 1;
    else
        return 0;
}

```

### SS of the OUTPUT

6. Write a C Program to Put Even & Odd Elements of an Array in 2 Separate Arrays.

#### CODE

```

#include<stdio.h>

int multiple(int, int);

int main()
{

```



```

int n, m;
printf("\nEnter any integer: ");
scanf("%d", &n);
printf("\nEnter an integer to check if it's a multiple of the previous number: ");
scanf("%d", &m);
printf("%d", multiple(n, m));
return 0;
}

int multiple(int n, int m)
{
    if(m%n == 0)
        return 1;
    else
        return 0;
}

```

## SS of the OUTPUT

7. Write a C Program to Cyclically Permute the Elements of an Array.

### CODE

```

#include<stdio.h>

int multiple(int, int);

int main()
{
    int n, m;

```

```
printf("\nEnter any integer: ");  
scanf("%d", &n);  
printf("\nEnter an integer to check if it's a multiple of the previous number: ");  
scanf("%d", &m);  
printf("%d", multiple(n, m));  
return 0;  
}  
  
int multiple(int n, int m)  
{  
    if(m%n == 0)  
        return 1;  
    else  
        return 0;  
}
```

**SS of the OUTPUT**

**8.**

**6.10** Use a single-subscripted array to solve the following problem. A company pays its salespeople on a commission basis. The salespeople receive \$200 per week plus 9 percent of their gross sales for that week. For example, a salesperson who grosses \$3000 in sales in a week receives \$200 plus 9 percent of \$3000, or a total of \$470. Write a C program (using an array of counters) that determines how many of the salespeople earned salaries in each of the following ranges (assume that each salesperson's salary is truncated to an integer amount):

1. \$200–\$299
2. \$300–\$399
3. \$400–\$499
4. \$500–\$599
5. \$600–\$699
6. \$700–\$799
7. \$800–\$899
8. \$900–\$999
9. \$1000 and over

## CODE

```
#include<stdio.h>

int multiple(int, int);

int main()
{
    int n, m;
    printf("\nEnter any integer: ");
    scanf("%d", &n);
    printf("\nEnter an integer to check if it's a multiple of the previous number: ");
    scanf("%d", &m);
    printf("%d", multiple(n, m));
    return 0;
}

int multiple(int n, int m)
{
    if(m%n == 0)
```

```
    return 1;  
    else  
        return 0;  
}
```

SS of the OUTPUT

9. Write a program which performs the following tasks:

- Initialize an integer array of 10 elements in main()
- Pass the entire array to a function modify()

-In modify()multiply each element of array by 3

-Return the control to main()and print the new array elements in main()

#### CODE

```
#include<stdio.h>

int multiple(int, int);

int main()
{
    int n, m;
    printf("\nEnter any integer: ");
    scanf("%d", &n);
    printf("\nEnter an integer to check if it's a multiple of the previous number: ");
    scanf("%d", &m);
    printf("%d", multiple(n, m));
    return 0;
}

int multiple(int n, int m)
{
    if(m%n == 0)
        return 1;
    else
        return 0;
}
```

#### SS of the OUTPUT

10. Write a program that interchanges the odd and even indexed components of an array.

For Example:

If input array is: 3, 5, 1, 8, 2, 9

The output array should be: 5, 3, 8, 1, 9, 2

#### CODE

```
#include<stdio.h>

int multiple(int, int);

int main()
{
    int n, m;
    printf("\nEnter any integer: ");
    scanf("%d", &n);
    printf("\nEnter an integer to check if it's a multiple of the previous number: ");
    scanf("%d", &m);
    printf("%d", multiple(n, m));
    return 0;
}

int multiple(int n, int m)
{
    if(m%n == 0)
        return 1;
    else
        return 0;
}
```

#### SS of the OUTPUT

11. Twenty-Five numbers are entered from the keyboard into an array. The number to be searched is entered through the keyboard by the user. Write a program to find if the number to be searched is present in the array and if it is present, display the number of times it appears in the array.

#### CODE

```

#include<stdio.h>

int multiple(int, int);

int main()
{
    int n, m;
    printf("\nEnter any integer: ");
    scanf("%d", &n);
    printf("\nEnter an integer to check if it's a multiple of the previous number: ");
    scanf("%d", &m);
    printf("%d", multiple(n, m));
    return 0;
}

int multiple(int n, int m)
{
    if(m%n == 0)
        return 1;
    else
        return 0;
}

```

### SS of the OUTPUT

12. Write a Program to delete all duplicate elements from an array. If a number is occurring more than 1 time, then all the duplicate entries must be deleted and the number must be present only 1 time in the array. Then display the resultant array.

### CODE

```

#include<stdio.h>

int multiple(int, int);

```

```

int main()
{
    int n, m;
    printf("\nEnter any integer: ");
    scanf("%d", &n);
    printf("\nEnter an integer to check if it's a multiple of the previous number: ");
    scanf("%d", &m);
    printf("%d", multiple(n, m));
    return 0;
}

int multiple(int n, int m)
{
    if(m%n == 0)
        return 1;
    else
        return 0;
}

```

### SS of the OUTPUT

13. Write a program to find whether the array of integers contains a duplicate number. If yes, then print the position of original number and all the duplicate numbers.

### CODE

```

#include<stdio.h>

int multiple(int, int);

int main()
{
    int n, m;

```



```

printf("\nEnter any integer: ");
scanf("%d", &n);
printf("\nEnter an integer to check if it's a multiple of the previous number: ");
scanf("%d", &m);
printf("%d", multiple(n, m));
return 0;
}
int multiple(int n, int m)
{
    if(m%n == 0)
        return 1;
    else
        return 0;
}

```

### SS of the OUTPUT

14. Write a program to find out a pair of two numbers in an array whose sum is closest to 0..

### CODE

```

#include<stdio.h>
int multiple(int, int);
int main()
{
    int n, m;
    printf("\nEnter any integer: ");
    scanf("%d", &n);

```

```
printf("\nEnter an integer to check if it's a multiple of the previous number: ");  
scanf("%d", &m);  
printf("%d", multiple(n, m));  
return 0;  
}  
  
int multiple(int n, int m)  
{  
    if(m%n == 0)  
        return 1;  
    else  
        return 0;  
}
```

**SS of the OUTPUT**