

Experiment No: 01

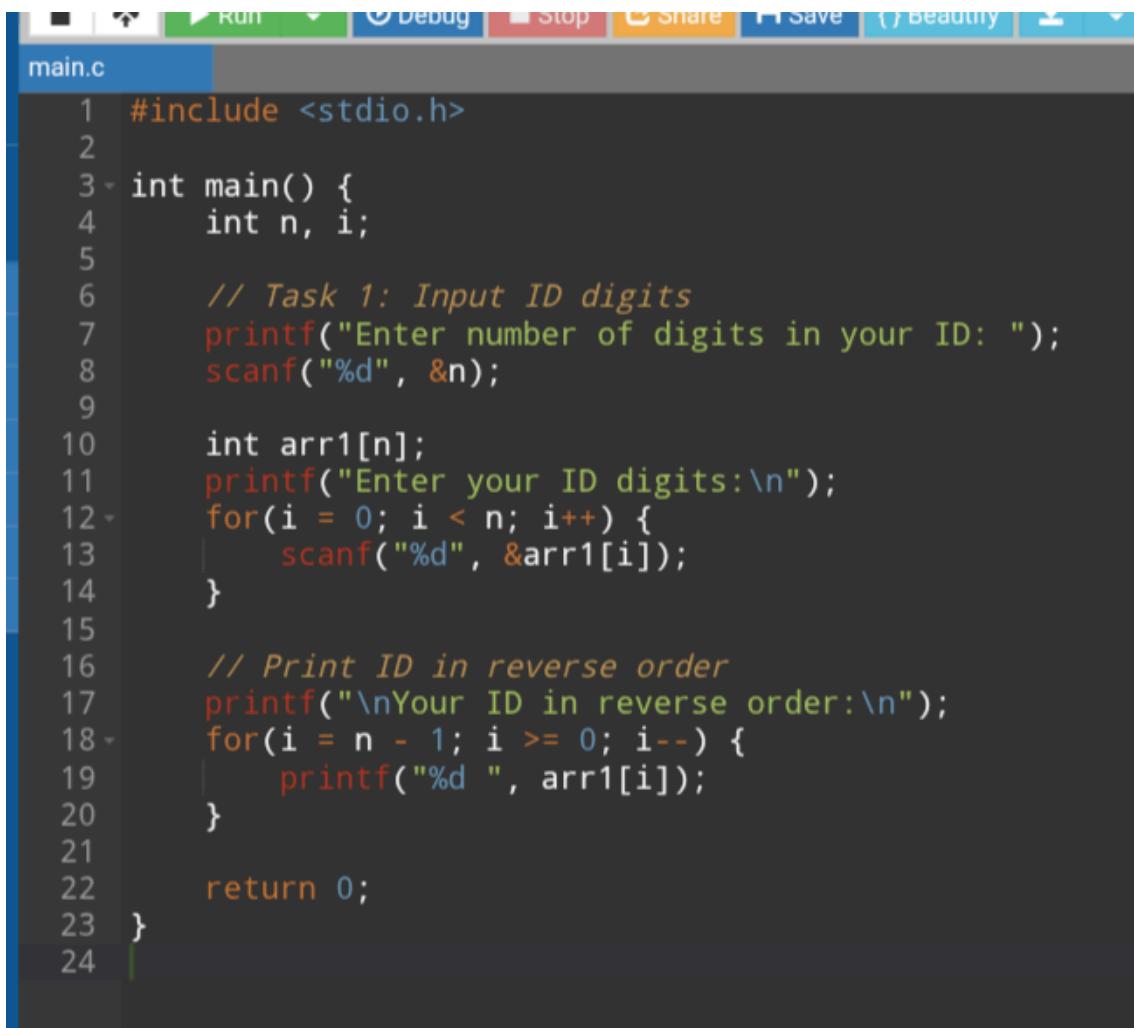
Experiment Name: Operations in Array

Objective: To learn how to store data in an array and display it in reverse order by traversing the array from end to start and performing multiplication

Task No: 01

Problem Statement: Input your own ID as an array and print the array in reverse order.

Source Code (C):



```

main.c
1 #include <stdio.h>
2
3 int main() {
4     int n, i;
5
6     // Task 1: Input ID digits
7     printf("Enter number of digits in your ID: ");
8     scanf("%d", &n);
9
10    int arr1[n];
11    printf("Enter your ID digits:\n");
12    for(i = 0; i < n; i++) {
13        scanf("%d", &arr1[i]);
14    }
15
16    // Print ID in reverse order
17    printf("\nYour ID in reverse order:\n");
18    for(i = n - 1; i >= 0; i--) {
19        printf("%d ", arr1[i]);
20    }
21
22    return 0;
23 }
24

```

Output:

```

input
Enter number of digits in your ID: 6
Enter your ID digits:
2 4 3 4 1 9

Your ID in reverse order:
9 1 4 3 4 2

...Program finished with exit code 0
Press ENTER to exit console.

```

Task No: 02

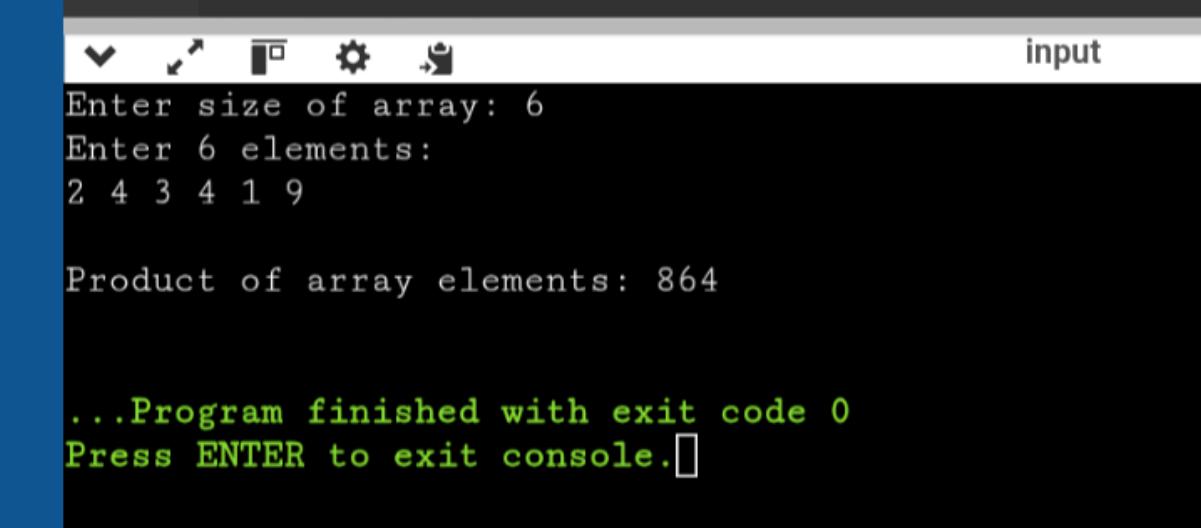
Problem Statement: Input an array from the user. Find the product of the elements of the array.

Source Code (C):

```

main.c
1 #include <stdio.h>
2
3 int main() {
4     int n, i;
5     long long product = 1;
6
7     // Task 2: Input array
8     printf("Enter size of array: ");
9     scanf("%d", &n);
10
11    int arr2[n];
12    printf("Enter %d elements:\n", n);
13    for(i = 0; i < n; i++) {
14        scanf("%d", &arr2[i]);
15    }
16
17    // Calculate product
18    for(i = 0; i < n; i++) {
19        product *= arr2[i];
20    }
21
22    // Print result
23    printf("\nProduct of array elements: %lld\n", product);
24
25    return 0;
26 }

```

Output:

```
Enter size of array: 6
Enter 6 elements:
2 4 3 4 1 9

Product of array elements: 864

...Program finished with exit code 0
Press ENTER to exit console.
```

Discussion: In this experiment, I learned how to efficiently manipulate arrays through different operations. By printing my ID in reverse order, I gained a clear understanding of array indexing and traversing arrays backward. The second task, calculating the product of array elements, reinforced the importance of initializing a variable correctly and iterating through all elements without errors. These tasks demonstrated how arrays can store and process multiple values systematically. I also observed how proper handling of loops ensures accurate results, even with larger datasets. Overall, this experiment strengthened my logical thinking and practical skills in array operations, laying a strong foundation for more advanced programming challenges.