Assignment 8

DSA LAB

2029196

Adarsh Kumar

Q1 WAP to read an array of integers and search for an element using linear search.

```c
#include <stdio.h>

int main()
{
    int array[100], search, c, n, count = 0;

    printf("Enter number of elements in array: ");
    scanf("%d", &n);

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

    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);

    printf("Enter a number to search: ");
    scanf("%d", &search);

    for (c = 0; c < n; c++)
    {
        if (array[c] == search)
        {
            printf("%d is present at location %d.\n", search, c +
1);
            count++;
        }
    }
    if (count == 0)
        printf("%d isn't present in the array.\n", search);
    else
        printf("%d is present %d times in the array.\n", search,
count);

    return 0;
}
```

OUTPUT:-

```
Enter number of elements in array: 6
Enter 6 numbers: 1 2 2 3 5 6
Enter a number to search: 2
2 is present at location 2.
2 is present at location 3.
2 is present 2 times in the array.
```

Q2. WAP to read an array of integers and search for an element using binary search.

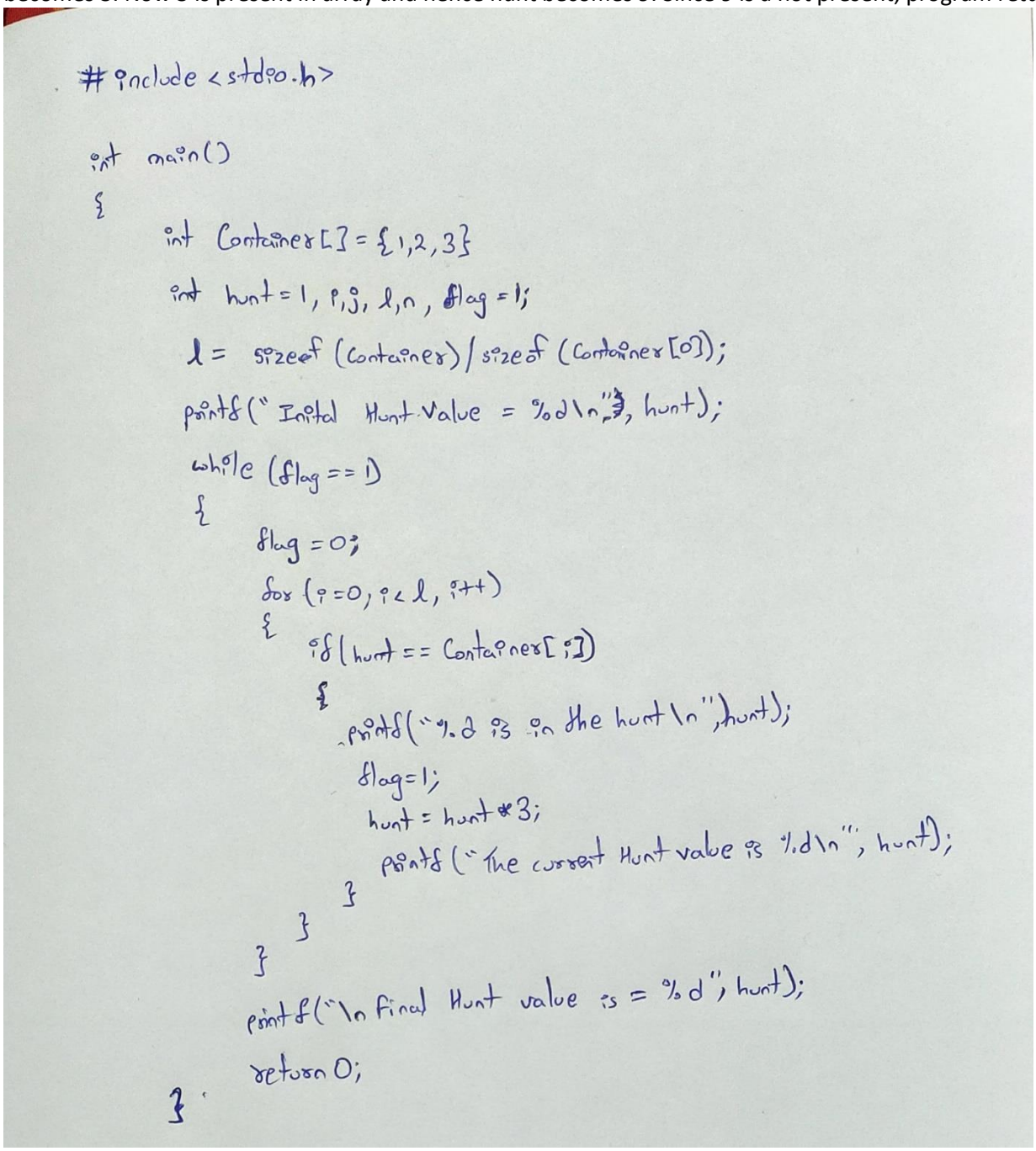
```c
#include <stdio.h>
int main()
{
    int c, first, last, middle, n, search, array[100];

    printf("Enter number of elements\n");
    scanf("%d", &n);

    printf("Enter %d integers\n", n);

    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);

    printf("Enter value to find\n");
    scanf("%d", &search);

    first = 0;
    last = n - 1;
    middle = (first + last) / 2;

    while (first <= last)
    {
        if (array[middle] < search)
            first = middle + 1;
        else if (array[middle] == search)
        {
            printf("%d found at location %d.\n", search, middle +
1);
            break;
        }
        else
            last = middle - 1;

        middle = (first + last) / 2;
    }
    if (first > last)
```

```
        printf("Not found! %d isn't present in the list.\n",
search);

    return 0;
}
```

OUTPUT:-

```
Enter number of elements
6
Enter 6 integers
1 3 5 7 9 4
Enter value to find
6
Not found! 6 isn't present in the list.
```

Q3 Given an array container and integer hunt. WAP to find whether hunt is present in container or not. If present, then triple the value of hunt and search again. Repeat these steps until hunt is not found. Finally return the value of hunt. Input: container = {1, 2, 3} and hunt = 1 then Output: 9 Explanation: Start with hunt = 1. Since it is present in array, it becomes 3. Now 3 is present in array and hence hunt becomes 9. Since 9 is a not present, program return 9.

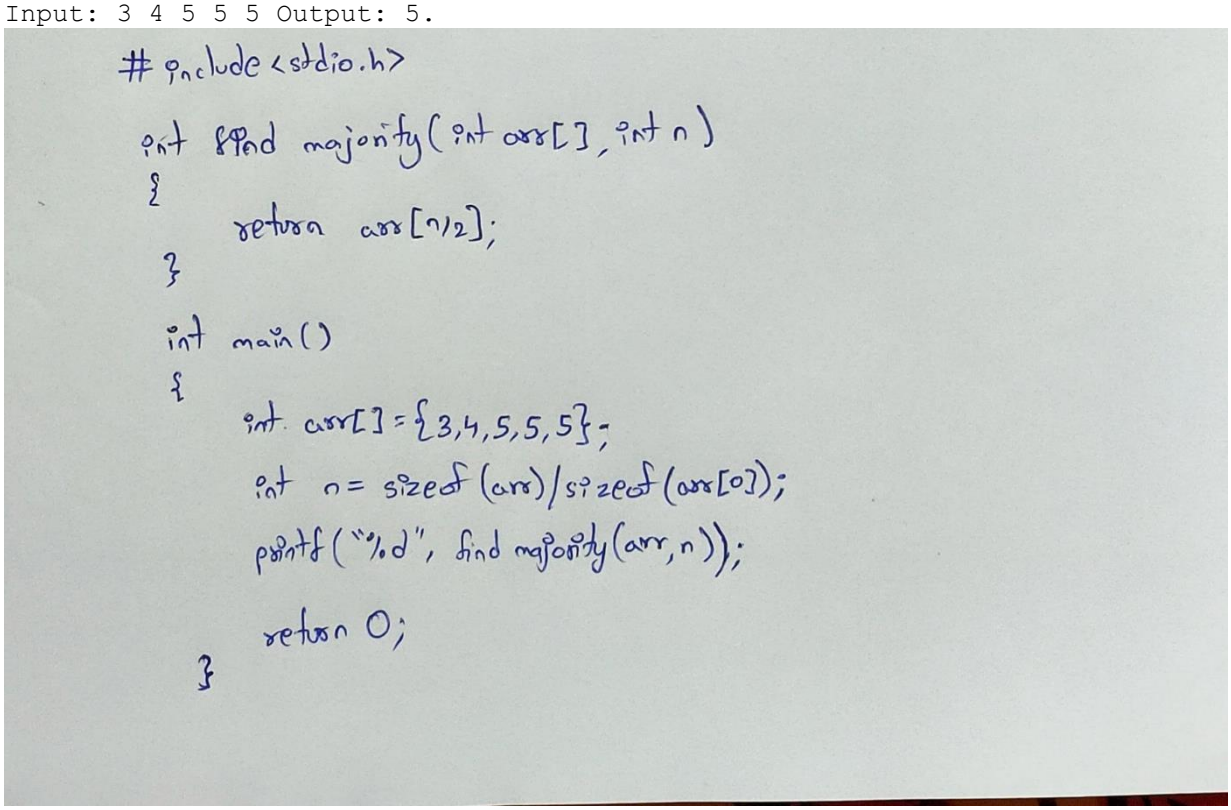


```
#include <stdio.h>

int main()
{
    int Container[] = {1,2,3}
    int hunt = 1, i,j, l,n, flag = 1;
     l = sizeof (Container) / sizeof (Container[0]);
     printf(" Inital Hunt Value = %d\n", hunt);
     while (flag == 1)
     {
         flag = 0;
         for (i=0; i<l, i++)
         {
            if(hunt == Container[i])
            {
              printf("%d is in the hunt\n",hunt);
               flag=1;
               hunt = hunt*3;
                printf(" The current Hunt value is %d\n", hunt);
            }
        }
    }
    printf("\n final Hunt value is = %d", hunt);
    return 0;
}
```

OUTPUT:-

```
Initial Hunt Value = 1
1 is in the hunt
The current Hunt value is: 3
3 is in the hunt
The current Hunt value is: 9

Final Hunt Value = 9
```

Q4. Given a sorted array of length n, WAP to find the number in array that appears more than or equal to n/2 times. It can be assumed that such element always exists.
Input: 2 3 3 4 Output: 3
Input: 3 4 5 5 5 Output: 5.



```
#include <stdio.h>

int find majority(int arr[], int n)
{
    return arr[n/2];
}

int main()
{
    int arr[] = {3,4,5,5,5};
    int n = sizeof(arr)/sizeof(arr[0]);
    printf("%d", find majority(arr,n));

    return 0;
}
```

OUTPUT:-

```
Desktop\3 rd
5
PS C:\Users\
```

Q5. WARP (Write a Recursive Program) to search an element in a dynamic array of n integers using linear search.

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

int main()
{
    int *array, search, c, n, count = 0;

    printf("Enter number of elements in array\n");
    scanf("%d", &n);

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

    for (c = 0; c < n; c++)
```

```
    {
        array = (int *)malloc(sizeof(int));
        scanf("%d", &array[c]);
    }

    printf("Enter a number to search\n");
    scanf("%d", &search);

    for (c = 0; c < n; c++)
    {
        if (array[c] == search)
        {
            printf("%d is present at location %d.\n", search, c +
1);
            count++;
        }
    }
    if (count == 0)
        printf("%d isn't present in the array.\n", search);
    else
        printf("%d is present %d times in the array.\n", search,
count);

    return 0;
}
```

OUTPUT:-

```
Enter number of elements in array: 6
Enter 6 numbers: 1 2 2 3 5 6
Enter a number to search: 2
2 is present at location 2.
2 is present at location 3.
2 is present 2 times in the array.
```

Q6. WARP using recursion to search an element in a dynamic array of n integers using binary search.

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

int main()
{
    int c, first, last, middle, n, search, *array;

    printf("Enter number of elements\n");
    scanf("%d", &n);
```

```
    printf("Enter %d integers\n", n);

    for (c = 0; c < n; c++)
    {
        array = malloc(sizeof(int));
        scanf("%d", &array[c]);
    }

    printf("Enter value to find\n");
    scanf("%d", &search);

    first = 0;
    last = n - 1;
    middle = (first + last) / 2;

    while (first <= last)
    {
        if (array[middle] < search)
            first = middle + 1;
        else if (array[middle] == search)
        {
            printf("%d found at location %d.\n", search, middle +
1);
            break;
        }
        else
            last = middle - 1;

        middle = (first + last) / 2;
    }
    if (first > last)
        printf("Not found! %d isn't present in the list.\n",
search);

    return 0;
}
```

OUTPUT:-