

# Programming Projects

1. Ans:

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

```
int binarySearch(int arr[], int l, int r, int x)
{
    if (r >= l) {
        int mid = l + (r - l) / 2;

        if (arr[mid] == x)
            return mid;

        else if (arr[mid] > x)
            return binarySearch(arr, l, mid - 1, x);
        else
            return binarySearch(arr, mid + 1, r, x);
    }

    return -1;
}

int main(void)
{
    int i, n, x, result;
    printf("Enter size of the array: ");
    scanf("%d", &n);

    int A[n];
    printf("Enter elements of the array (in ascending order) : ");
    for(i = 0; i < n; i++)
        scanf("%d", &A[i]);

    printf("Enter number to be searched: ");
    scanf("%d", &x);

    result = binarySearch(A, 0, n - 1, x);
    (result == -1)
        ? printf("Number is not present in array")
        : printf("Number is present at index %d", result);

    return 0;
}
```

**O/P**

**Enter size of the array: 4**  
**Enter elements of the array (in ascending order) : 1 3 5 7**  
**Enter number to be searched: 6**  
**Number is not present in array**

**Enter size of the array: 4**  
**Enter elements of the array (in ascending order) : 1 3 5 7**  
**Enter number to be searched: 1**  
**Number is present at index 0**

2. Ans:

```
#include <stdio.h>
void bubble_sort(int[], int);
void main()
{
    int n, i;
    printf("Enter size of the array: ");
    scanf("%d", &n);

    int A[n];
    printf("Enter elements of the array: ");
    for(i = 0; i < n; i++)
        scanf("%d", &A[i]);

    printf("Before Sorting : ");
    for(i = 0; i < n; i++)
        printf("%d ", A[i]);

    bubble_sort(A, n);
    printf("\nAfter Sorting : ");
    for(i = 0; i < n; i++)
        printf("%d ", A[i]);

    printf("\n");
}

void bubble_sort(int A[], int n)
{
    int i, j, temp;

    for(i = 0; i < (n - 1); i++)
    {
        for(j = 0; j < (n - i - 1); j++)
        {
            if(A[j] > A[j + 1])
            {
                temp = A[j];
                A[j] = A[j + 1];
                A[j + 1] = temp;
            }
        }
    }
}
```

**Q/P**

**Enter size of the array: 4**  
**Enter elements of the array: 5 3 7 1**  
**Before Sorting : 5 3 7 1**  
**After Sorting : 1 3 5 7**

3. Ans:

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

```

int hydroxide(char[]);

void main()
{
    char chemical[8];
    printf("\nEnter the name of chemical compound - ");
    scanf("%s", chemical);
    if(hydroxide(chemical))
        printf("\n%s is a hydroxide.",chemical);
    else
        printf("\n%s is not a hydroxide.",chemical);
}

int hydroxide(char chemical[])
{
    char *ptr = strstr(chemical, "OH");

    if (ptr != NULL) /* Substring found */
        return 1;
    else /* Substring not found */
        return 0;
}

```

#### **O/P**

**Enter the name of chemical compound - KOH**

**KOH is a hydroxide.**

**Enter the name of chemical compound - H2O2**

**H2O2 is not a hydroxide.**

**Enter the name of chemical compound - KOHCL**

**KOHCL is not a hydroxide.**

4. Ans:

```

#include <stdio.h>
#include <string.h>

void main()
{
    char noun[20], plural[20];
    printf("\nEnter the noun - ");
    scanf("%s", noun);
    if(noun[strlen(noun)-1]=='y'){
        strncpy(plural,noun,strlen(noun)-1);
        strcat(plural,"ies");
        printf("\nPlural of %s is %s.",noun,plural);}
    else if(noun[strlen(noun)-1]=='s' || (noun[strlen(noun)-2]=='c' && noun[strlen(noun)-1]=='h')
|| (noun[strlen(noun)-2]=='s' && noun[strlen(noun)-1]=='h'))){
        strcpy(plural,noun);
        strcat(plural,"es");
        printf("\nPlural of %s is %s.",noun,plural);}
    else {
        strcpy(plural,noun);
        strcat(plural,"s");
    }
}

```

```
        printf("\nPlural of %s is %s.",noun,plural);}
}
```

**O/P**

**Enter the noun - fly**

**Plural of fly is flies.**

**Enter the noun - boss**

**Plural of boss is bosses.**

**Enter the noun - church**

**Plural of church is churches.**

**Enter the noun - dish**

**Plural of dish is dishes.**

**Enter the noun - dog**

**Plural of dog is dogs.**