

<b>Name</b>	Jhaveri Varun Nimitt
<b>UID no.</b>	2023800042
<b>Experiment No.</b>	4

<b>AIM:</b>	Demonstrate the use of one-dimensional arrays to solve a given problem.
<b>Program 1</b>	
<b>PROBLEM STATEMENT :</b>	Write a C Program which contains a function to perform search of a particular element on an array. Create an array in main() and call the function to test it.
<b>ALGORITHM:</b>	<ol style="list-style-type: none"> <li>1. Start</li> <li> <ul style="list-style-type: none"> <li>● 2. Define a void function named display that takes an integer arrays[] and an integer size</li> <li>● 3. Start a for loop from i=0 to size-1, incrementing i by 1 each iteration               <ol style="list-style-type: none"> <li>a. Print arrays[i] followed by a space</li> </ol> </li> <li>● 4. End for loop</li> <li>● 5. End function display</li> <li><input type="checkbox"/> 6. Define an int function named ispresent that takes an integer array arrayforpresent[], an integer arraysizes, and an integer whotocheck</li> <li><input type="checkbox"/> 7. Declare an integer flag and initialize it to 0</li> <li><input type="checkbox"/> 8. Start a for loop from i=0 to arraysizes-1, incrementing i by 1 each iteration               <ol style="list-style-type: none"> <li>a. Check if arrayforpresent[i] is equal to whotocheck                   <ol style="list-style-type: none"> <li>i. If it is, set flag to 1 and return 1</li> </ol> </li> </ol> </li> <li><input type="checkbox"/> 9. If flag is 0, return 0</li> <li><input type="checkbox"/> 10. End function ispresent</li> </ul> </li> <li>11. Start main function</li> <li>12. Declare an integer variable arrsize</li> </ol>

	<ol style="list-style-type: none"> <li>13. Print "Enter size:"</li> <li>14. Scan an integer into arrsize</li> <li>15. Declare an integer array arr of size arrsize</li> <li>16. Start a for loop from i=0 to arrsize-1, incrementing i by 1 each iteration <ol style="list-style-type: none"> <li>a. Print "Enter" (i+1) "element:"</li> <li>b. Scan an integer into arr[i]</li> </ol> </li> <li>17. Call the display function with arguments arr and arrsize</li> <li>18. Declare an integer variable checksum</li> <li>19. Print "Enter value which u want to check for:"</li> <li>20. Scan an integer into checksum</li> <li>21. If ispresent(arr, arrsize, checksum) ==1 <ol style="list-style-type: none"> <li>a. Print "yes,it is present"</li> </ol> </li> <li>22. Else <ol style="list-style-type: none"> <li>a. Print "no, it is not present"</li> </ol> </li> <li>23. End main function</li> <li>24. End</li> </ol>
<b>PROGRAM:</b>	<pre> #include &lt;stdio.h&gt; void display(int arrays[],int size); int ispresnt(int arrayforpresent[],int arraysiz, int whotocheck); int main() {     int arrsize;     printf("Enter size :");     scanf("%d",&amp;arrsize);     int arr[arrsize];     for(int i =0;i&lt;arrsize;i++)     {         printf("Enter %d element:",(i+1));         scanf("%d",&amp;arr[i]);     } } </pre>

```

    }
    display(arr, arrsize);
    int checksum;
    printf("\nEnter value which u want to check for:");
    scanf("%d", &checksum);
    if (ispresnt(arr, arrsize, checksum) == 1)
    {
        printf("\nyes, it is present");
    }
    else
    {
        printf("\n no, it is not present");
    }
}

void display(int arrays[], int size)
{
    for(int i=0; i<size; i++)
    {
        printf("%d ", arrays[i]);
    }
}

int ispresnt(int arrayforpresent[], int arraysize, int whotocheck)
{
    int flag = 0;
    for(int i=0; i<arraysize; i++)
    {
        if (arrayforpresent[i] == whotocheck)
        {
            flag = 1;
            return 1;
        }
    }
    if(flag == 0)
    {
        return 0;
    }
}

```

```
}
```

**RESULT:**

```
cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 4$ gcc array4.c
cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 4$ ./a.out
Enter size :5
Enter 1 element:98
Enter 2 element:423
Enter 3 element:23
Enter 4 element:88
Enter 5 element:5
98 423 23 88 5
Enter value which u want to check for:5

yescyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 4$
```

**Program 2****PROBLEM  
STATEMENT :**

Write a C Program which contains a function to sort an array using selection sort. Create an array in main() and call the function to test it.

**ALGORITHM:**

1. Start
- 2. Define a void function named sort that takes an integer arrays[] and an integer size
- 3. Declare integer variables i, j, and min\_index initializing them all to 0
- 4. Start a for loop from i=0 to size, incrementing i by 1 each iteration.
  - a. Set min\_index to i
  - b. Start a nested for loop from j=i+1 to size, incrementing j by 1 each iteration.
    - i. Check if arrays[j] is less than arrays[min\_index]  
- If true, set min\_index to j
  - c. Declare an integer variable temp and initialize it to 0
  - d. Set temp to arrays[min\_index]
  - e. Swap arrays[min\_index] with arrays[i]
  - f. Swap arrays[i] with temp
- 5. Start a for loop from i=0 to size-1, incrementing i by 1 each iteration.
  - a. Print "New array :" followed by arrays[i] with a new line

- 6. End function sort

7. Start main function

8. Declare an integer variable arrsize

9. Print "Enter size:"

10. Scan an integer into arrsize

11. Declare an integer array arr of size arrsize

12. Start a for loop from i=0 to arrsize-1

- a. Print "Enter" (i+1) "element:"
- b. Scan an integer into arr[i]

13. Call the sort function with arguments arr and arrsize

14. End main function

15. End

**PROGRAM:**

```
#include <stdio.h>
void sort(int arrays[],int size);

int main()
{
    int arrsize;
    printf("Enter size :");
    scanf("%d",&arrsize);
    int arr[arrsize];
    for(int i =0;i<arrsize;i++)
    {
        printf("Enter %d element:",(i+1));
        scanf("%d",&arr[i]);
    }
    sort(arr,arrsize);
}

void sort(int arrays[],int size)
```

```
{  
    int i, j, min_index=0;  
    for (i = 0; i < size; i++)  
    {  
        min_index = i;  
        for (j = i+1; j < size; j++)  
        {  
            if (arrays[j] < arrays[min_index])  
            {  
                min_index = j;  
            }  
        }  
        int temp=0;  
        temp= arrays[min_index];  
        arrays[min_index] = arrays[i];  
        arrays[i] = temp;  
    }  
    for(int i =0;i<size;i++)  
    {  
        printf("\nNew array : %d",arrays[i]);  
    }  
}
```

**RESULT:**

```
cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 4$ gcc selectionsorts.c
cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 4$ ./a.out
Enter size :5
Enter 1 element:48
Enter 2 element:99999
Enter 3 element:43823
Enter 4 element:
1394
Enter 5 element:2

New array : 2
New array : 48
New array : 1394
New array : 43823
New array : 99999cyclops@cyclops:~/Desktop/PSIPL Semester 1/Experiment 4$
```

**CONCLUSION:**

**I have understood the way to use one-dimensional arrays to solve a given problem**