One Dimensional Numeric Arrays – Practice Problems
What are the different ways to initialize the array values?
What are the different ways to initialize the array values?

```c
int a[10] = {0, 1, 4, 6, 7, 12, 8, 9, -1, 34};

OR

for (i = 0; i < 10; i++)
    scanf("%d", &a[i]);
```
Practice Problems

Execute them during lab hours
Problem 1

• Consider array of 20 integers A[20]={1, 5, 4, 8, 9, 2, 0, 5, 1, 7, 8, 3, 1, 8, 9, 5, 3, 9, 4, 2}. User enters the value which will be searched through in an array A. Depending on if the value is found or not found, display the appropriate message.
Problem 1 Solution

for (i=0; i<20; i++)
    if (A[i] == user_input)
    {
        printf("%d found",user_input);
        break;
    }
if(i==20)
    printf("%d not found",user_input);
Problem 2

• Problem 2: Consider array of 20 integers
  \( A[20] = \{1, 5, 4, 8, 9, 2, 0, 5, 1, 7, 8, 3, 1, 8, 9, 5, 3, 9, 4, 2\} \). Reverse the order of the element
  \( A[20] = \{2, 4, 9, 3, 5, \ldots \} \)
Problem 2 Solution

for (i=0; i<20/2; i++)
{
    temp = A[i];
    A[19-i] = temp;
}

Problem 3 and 4

• In an array with 10 float values, find the largest and smallest of all elements

• In an array with 10 float values, find the second largest value.
Problem 3 Solution

int max = -99999, min=99999;
for (i=0; i<20; i++)
{
    if (max < A[i])
        max = A[i];
    if (min > A[i])
        min = A[i];
}
Problem 4 Solution

```c
int max = -99999, sec_max = -99999;
for (i=0; i<20; i++)
{
    if (max < A[i])
    {
        sec_max = max;
        max = A[i];
    }
    else if (sec_max < A[i] && max != A[i])
    {
        sec_max = A[i];
    }
}
```
Problem 5

• In an array with 10 integer values, assuming that array is sorted in ascending order, perform binary search to find the value entered by user
What is binary search?

If searching for 23 in the 10-element array:

2  5  8  12  16  23  38  56  72  91

23 > 16, take 2\textsuperscript{nd} half

2  5  8  12 \textcolor{blue}{16}  23  38  56  72  91

23 < 56, take 1\textsuperscript{st} half

2  5  8  12  16  23  38  \textcolor{blue}{56}  72  91

Found 23, Return 5

2  5  8  12  16  23  38  \textcolor{blue}{23}  56  72  91
while (1) {
    mid = low + ((high - low)/2);
    printf("low=%d, mid=%d, high=%d\n", low, mid, high);
    if (A[mid] == search_val) {
        printf("value %d is found", search_val);
        break;
    } else if (A[mid] > search_val) {
        high = mid-1;
    } else if (A[mid] < search_val) {
        low = mid+1;
    }
    if (low > high) {
        printf("value %d not found", search_val);
        break;
    }
}