

Assignment 5: Arrays

Garvit Shah

F24

U21CS089

1. WAP to add corresponding elements of two 1-Dimensional arrays and store in the third array, also calculate the average of the third array.

```
//Q1
#include <stdio.h>
int main(){
    int m;
    float sum=0;
    printf("Enter the length of the arrays - ");
    scanf("%d", &m);
    int n1[m], n2[m], n3[m], i=0;
    printf("Enter the first array - \n");
    for (i=0; i<m; i++){
        scanf("%d", &n1[i]);
    }

    printf("Enter the second array - \n");
    for (i=0; i<m; i++){
        scanf("%d", &n2[i]);
    }
    i=0;
    for (i=0; i<m; i++){
        n3[i] = n1[i] + n2[i];
    }

    printf("Addition is - ");
    for (i=0; i<m; i++){
        printf("%d ", n3[i]);
        sum += n3[i];
    }
    printf("\nAverage is %f", sum/m);
}
```

```

//Q1/3/garvishan/desktop/college/1st/Assignment1
Enter the length of the arrays - 5
Enter the first array -
1 2 3 4 5
Enter the second array -
2 3 1 4 5
Addition is - 3 5 4 8 10
Average is 6.000000

```

2. WAP to sort an array in descending order.

```

//Q2
#include <stdio.h>
int main(){
    int m;
    printf("Enter the length of the array - ");
    scanf("%d", &m);
    int arr1[m], i, j, k, a;
    printf("Enter the array - \n");
    for (i=0;i<m;i++){
        scanf("%d", &arr1[i]);
    }
    for(a=1;a<m;a++){
        for (i=0; i<m-1; i++){
            if (arr1[i]<arr1[i+1]){
                j = arr1[i];
                arr1[i] = arr1[i+1];
                arr1[i+1] = j;
            }
        }
    }
    printf("Sorted Array - ");
    for (k=0;k<m;k++){
        printf("%d ", arr1[k]);
    }
    return 0;
}
//Q2/3/garvishan/desktop/college/1st/Assignment1
Enter the length of the array - 5
Enter the array -
1 3 2 5 6
Sorted Array - 6 5 3 2 1

```

3. WAP to count total no of odd and even numbers from the 1-D array.

//Q3

```
#include <stdio.h>
int main(){
    int m;
    printf("\nEnter the length of the array - ");
    scanf("%d", &m);
    int e=0, o=0, arr[m], i;
    printf("Enter the array - \n");
    for (i = 0; i<m; i++){
        scanf("%d", &arr[i]);
    }
    for (i=0; i<m;i++){
        if ((arr[i]%2) == 0){
            e += 1;
        }
        else{
            o+=1;
        }
    }
    printf("\nThe array is\n>> ");
    for (i = 0; i<m ; i++){
        printf("%d ", arr[i]);
    }
    printf("\n\nNo. of Even Numbers are - %d\n", e);
    printf("No. of Odd Numbers are - %d\n", o);
    return 0;
}
```

Enter the length of the array - 5

Enter the array -

1 2 4 5 6

The array is

>> 1 2 4 5 6

No. of Even Numbers are - 3

No. of Odd Numbers are - 2

-

4. WAP to exchange the smallest and largest values in 1-D array.

//Q4

```
#include <stdio.h>
int main() {
    int m, k, i, j;
    printf("\nEnter the length of the array - ");
    scanf("%d", &m);
    int arr[m], arr1[m];
    printf("Enter the array - \n");
    for (i=0; i<m; i++){
        scanf("%d", &arr[i]);
        arr1[i] = arr[i];
    }

    for(k=1; k<m; k++){
        for (i=0; i<m-1; i++){
            if (arr[i]<arr[i+1]){
                j = arr[i];
                arr[i] = arr[i+1];
                arr[i+1] = j;
            }
        }
    }

    for (i=0; i<m; i++){
        if (arr1[i] == arr[0]){
            arr1[i] = arr[m-1];
        }
        else if (arr1[i] == arr[m-1]){
            arr1[i] = arr[0];
        }
    }
    printf("\nArray with values exchanged - \n");
    for (i = 0; i<m; i++){
        printf("%d ", arr1[i]);
    }
    printf("\n");
    return 0;
}
```

```
Enter the length of the array - 5
Enter the array -
1 3 9 -10 4
```

```
Array with values exchanged -
1 3 -10 9 4
```

5. WAP to delete an element of an array given by the user.

//Q5

```
#include <stdio.h>
int main(){
    int m;
    printf("\nEnter the length of the array - ");
    scanf("%d", &m);

    int arr[m], ele, i, arr1[m-1], k;
    printf("Enter the array - \n");
    for (i=0; i<m; i++){
        scanf("%d", &arr[i]);
    }

    printf("\nEnter the value to be deleted - ");
    scanf("%d", &ele);
    for (i=0; i<m; i++){
        if (ele == arr[i]){
            k = i;
            break;
        }
    }

    for (i = 0; i<m; i++){
        if (i<k){
            arr1[i] = arr[i];
        }
        else{
            arr1[i] = arr[i+1];
        }
    }
}
```

```

}

printf("\n");

for (i = 0; i<m-1; i++){
    printf("%d ", arr1[i]);
}

return 0;
}

```

```

Enter the length of the array - 5
Enter the array -
1 3 6 2 7

Enter the value to be deleted - 3

1 6 2 7

```

6. WAP to insert an element in an array specified by the user.

```

//Q6
#include <stdio.h>
int main(){
    int m;
    printf("Enter the length of the array - ");
    scanf("%d", &m);

    int arr[m], ele, i, arr1[m-1], k;
    printf("Enter the array - \n");
    for (i=0; i<m;i++){
        scanf("%d", &arr[i]);
    }

    printf("\nEnter the value to be inserted - ");
    scanf("%d", &ele);
    printf("Enter the position - ");
    scanf("%d", &k);
}

```

```

for (i = 0; i<m+1; i++){
    if (i<(k-1)){
        arr1[i] = arr[i];
    }
    else if (i == (k-1)){
        arr1[i] = ele;
    }
    else{
        arr1[i] = arr[i-1];
    }
}
printf("\n");
for (i = 0; i<m+1; i++){
    printf("%d ", arr1[i]);
}

return 0;
}

```

`//user3/garvitsdhan/desktop/college/1st/Assignments`

Enter the length of the array - 5

Enter the array -

1 3 2 4 5

Enter the value to be inserted - 6

Enter the position - 3

1 3 6 2 4 5

7. Given an array `arr[]` of size `N`. The task is to find the sum of `arr[i] % arr[j]` for all valid pairs. Answer can be large. So, output answer modulo 1000000007 Input: `arr[] = {1, 2, 3}`
Output: 5
 $(1 \% 1) + (1 \% 2) + (1 \% 3) + (2 \% 1) + (2 \% 2) + (2 \% 3) + (3 \% 1) + (3 \% 2) + (3 \% 3) = 5$

`//Q7`

`#include <stdio.h>`

`int main(){`

`int m;`

`printf("\nEnter the length of the array - ");`

```

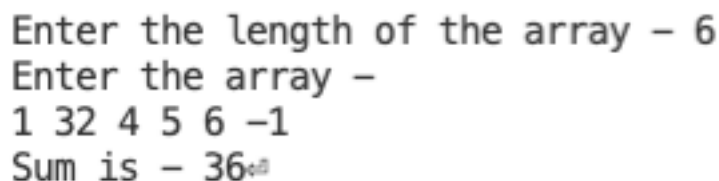
scanf("%d", &m);

int arr[m], i, j, sum=0;
printf("Enter the array - \n");
for (i = 0; i<m; i++){
    scanf("%d", &arr[i]);
}
for(i=0; i<m; i++){
    for (j=0; j<m; j++){
        if (arr[j] != 0){
            sum += arr[i]*arr[j];
        }
    }
}

printf("Sum is - %d", sum);

return 0;
}

```



```

Enter the length of the array - 6
Enter the array -
1 32 4 5 6 -1
Sum is - 36

```

8. WAP to perform matrix multiplication of 3*3 matrixes

```

//Q8
#include <stdio.h>
int main(){
    int M1[3][3], M2[3][3], M3[3][3], i, j, k;
    printf("Enter the 1st matrix - \n");
    for (i=0; i<3; i++){
        printf("Enter the row-%d : ", i+1);
        for (j=0; j<3; j++){
            scanf("%d", &M1[i][j]);
        }
    }
}

```



```

}
printf("\n\n");
printf("Enter the 2nd matrix - \n");
for (i=0; i<3; i++){
    printf("Enter the row-%d : ", i+1);
    for (j=0; j<3; j++){
        scanf("%d", &M2[i][j]);
    }
}
}

```

```

for(i = 0; i<3; i++){
    printf("Row-%d : ", i+1);
    for (j = 0; j<3; j++){
        int sum= 0;
        for(k=0; k<3; k++){
            sum += M1[i][k]*M2[k][j];
        }
        printf("%d ", sum);
    }
    printf("\n");
}
return 0;
}

```

/usr/bin/gcc -g -lm -o desktop/callege/101/100

```

Enter the 1st matrix -
Enter the row-1 : 1 1 0
Enter the row-2 : 1 2 2
Enter the row-3 : 1 3 2

```

```

Enter the 2nd matrix -
Enter the row-1 : 2 3 1
Enter the row-2 : 2 0 0
Enter the row-3 : 3 1 1
Row-1 : 4 3 1
Row-2 : 12 5 3
Row-3 : 14 5 3

```

9. Given an array of integers of size n, find out if the numbers in the array appear in a palindromic order. A palindrome is a sequence that reads the same when you flip it. For example, 121 is a palindrome, 3 is a palindrome, and 234432 is also a palindrome

//Q9

```
#include <stdio.h>
int main(){
    int m;
    printf("\nEnter the length - ");
    scanf("%d", &m);
    int arr[m], x=0, i;
    printf("Enter the array -\n");
    for (i = 0; i<m; i++){
        scanf("%d", &arr[i]);
    }

    for (i = 0; i<m; i++){
        if (arr[i] == arr[m-1-i]){
            x+=1;
        }
    }

    if (x == m){
        for (i = 0; i<m; i++){
            printf("%d ", arr[i]);
        }
        printf(" is a Palindromic Array");
    }
    else{
        for (i = 0; i<m; i++){
            printf("%d ", arr[i]);
        }
        printf(" Not Palindromic Array");
    }
}
```

```

Enter the length - 5
Enter the array -
1 3 2 4 5
1 3 2 4 5 Not Palindromic Array
garvitshah@Garvits-MacBook-Air ~/D/C/FCP> cd "/Users/garvitshah/Desktop/College/FCP/Assignment5"

```

```

Enter the length - 4
Enter the array -
1 2 2 1
1 2 2 1 is a Palindromic Array

```

10. Given two sorted arrays of sizes m and n , write a program that merges the two into another array of size $m + n$ such that this new array also remains sorted.

```

//Q10
#include <stdio.h>
int main(){
    int array1[50], array2[50], array3[100], m, n, i, j, k = 0;
    printf("\n Enter size of array Array 1: ");
    scanf("%d", &m);
    printf("\n Enter sorted elements of array 1: \n");
    for (i = 0; i < m; i++){
        scanf("%d", &array1[i]);
    }
    printf("\n Enter size of array 2: ");
    scanf("%d", &n);
    printf("\n Enter sorted elements of array 2: \n");
    for (i = 0; i < n; i++){
        scanf("%d", &array2[i]);
    }
    i = 0; j = 0;
    while (i < m && j < n){
        if (array1[i] < array2[j]){
            array3[k] = array1[i];
            i++;
        }
        else{
            array3[k] = array2[j];
            j++;
        }
    }
}

```

```

        }
        k++;
    }
    if (i >= m){
        while (j < n){
            array3[k] = array2[j];
            j++;
            k++;
        }
    }
    if (j >= n){
        while (i < m){
            array3[k] = array1[i]; i++; k++;
        }
    }
    printf("\n After merging: \n");
    for (i = 0; i < m + n; i++){
        printf("%d ", array3[i]);
    }
}

```

Enter size of array Array 1: 4

Enter sorted elements of array 1:
1 2 3 4

Enter size of array 2: 3

Enter sorted elements of array 2:
5 6 7

After merging:

1
2
3
4
5
6
7

11. WAP to subtract 2-D Matrices.

```
//Q11
#include <stdio.h>
int main(){
    int M1[3][3], M2[3][3], M3[3][3], i,j;
    printf("Enter the 1st Matrix - \n");
    for(i = 0; i<3; i++){
        printf("Row - %d : ", i+1);
        for(j=0;j<3;j++){
            scanf("%d", &M1[i][j]);
        }
    }
    printf("Enter the 2nd Matrix - \n");
    for(i = 0; i<3; i++){
        printf("Row - %d : ", i+1);
        for(j=0;j<3;j++){
            scanf("%d", &M2[i][j]);
            M3[i][j] = M1[i][j] - M2[i][j];
        }
    }
    printf("Enter the resultant matrix - \n");
    for(i=0;i<3;i++){
        printf("Row - %d : ", i+1);
        for(j=0;j<3;j++){
            printf("%d ", M3[i][j]);
        }
        printf("\n");
    }
}
```

Enter the 1st Matrix -
Row - 1 : 1 2 1
Row - 2 : 2 0 1
Row - 3 : 1 3 2
Enter the 2nd Matrix -
Row - 1 : 0 1 3
Row - 2 : 1 2 4
Row - 3 : 1 1 0
Enter the resultant matrix -
Row - 1 : 1 1 -2
Row - 2 : 1 -2 -3
Row - 3 : 0 2 2