Assignment - 7

(1)
```
#include <stdio.h>
void main()
{
  int i, n, a[100];
  printf("\n\n Read a number of values in an
          array and display it in reverse
          order : \n");
  printf("-----------------------\n");
  printf("Input the number of elements to
          store in the array :");
  scanf("%d", &n);
  printf("Input %d number of elements in
          the array :\n", n);
  for(i=0; i<n; i++)
  {
    printf("element - %d :", i);
    scanf("%d", &a[i]);
  }
  printf("\n The values store into the array
          are : \n");
  for(i=0; i<n; i++)
  {
    printf("%5d", a[i]);
  }
  printf("\n\n The values store into the
          array in reverse are : \n");
  for(i=n-1; i>=0; i--)
  {
    printf("%5d", a[i]);
  }
  printf("\n\n");
}
```

output :

Read n number of values in an array and display it in reverse order :

Input the number of elements to store in the array : 3

Input 3 number of elements in the array :

element - 0 : 2

element - 1 : 5

element - 2 : 7

The values store into the array are :

2 5 7

The values store into the array in reverse are :

7 5 2

(2)

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a[1000], i, n, sum = 0;
    printf("Enter size of the array : ");
    scanf("%d", &n);
    printf("Enter elements in array : ");
    for(i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }
    for(i = 0; i < n; i++)
    {
        sum += a[i];
    }
    printf("Sum of array is : %d", sum);
    return 0;
}
```

Output :

Enter size of the array : 5

Enter elements in array : 7
8
9
4
5

Sum of an array is : 33

(3)

```
#include <stdio.h>
int main()
{
  int arr1[] = {1,2,3,4,5};
  int length = sizeof(arr1)/sizeof(arr1[0]);
  int arr2[length];
  for (int i=0; i<length; i++)
  {
    arr2[i] = arr1[i];
  }
  printf("Elements of original array:\n");
  for (int i=0; i<length; i++)
  {
    printf("%d", arr1[i]);
  }
  printf("\n");
  printf("Elements of new array:\n");
  for (int i=0; i<length; i++)
  {
    printf("%d", arr2[i]);
  }
return 0;
}
```

Elements of original array:
1 2 3 4 5
Elements of new array:
1 2 3 4 5.

(4)
```
#include <stdio.h>
void main()
{
  int arr1 [100];
  int arr2 [100];
  int arr3 [100];
  int n, num = 1, ctr = 0;
  int i, j;
  printf ("\n\n Count total number of
          duplicate elements in an array:\n");
  printf ("..........\n");
printf ("Input the number of elements to
          be stored in the array:");
scanf ("%d", &n);
printf ("Input %d elements in the array:\n",
                                    n);
for ( i = 0; i < n; i++)
{
  printf ("element - %d: ", i);
  scanf ("%d", &arr1[i]);
}
for (i = 0; i < n; i++)
{
  arr2 [i] = arr1[i];
  arr3 [i] = 0;
}
for (i = 0; i < n; i++)
{
  for (j = 0; j < n; j++)
```

```
    {
      if (arr1[i] == arr2[i])
      {
        arr3[i] = mm;
        mm++;
      }
    }
    mm = 1;
  }
  For (i=0; i<n; i++)
  {
    if (arr3[i] == 2) { ctr++; }
  }
  printf ("The total number of duplicate
          elements found in the array is: %d
          \n", ctr);
  printf ("\n \n");
}
```

output:

Count total number of duplicate elements in an array:

Input the number of elements to be stored in the array: 3
Input 3 elements in the array:
element - 0: 5
element - 1: 1
element - 2: 1
Total number of duplicate elements found in the array is: 1

(5)
```
#include <stdio.h>
#define MAX_SIZE 100
int main()
{
    int arr[MAX_SIZE];
    int i, max, min, size;
    printf("Enter size of the array: ");
    scanf("%d", &size);
    printf("Enter elements in the array: ");
    for(i=0; i<size; i++)
    {
        scanf("%d", &arr[i]);
    }
    max = arr[0];
    min = arr[0];
    for(i=1; i<size; i++)
    {
        if(arr[i] > max)
        {
            max = arr[i];
        }
        if(arr[i] < min)
        {
            min = arr[i];
        }
    }
    printf("maximum element = %d\n", max);
    printf("minimum element = %d\n", min);
    return 0;
}
```

Output :

Enter size of the array : 10.

Enter elements in the array : -10 10 0 -2 50 100 20 -1 10.

Maximum elements : 100.

Minimum elements : -10.

(6)

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
  int arr[100], i, num;
  printf("Enter size of the array\n");
  scanf("%d", &num);
  printf("Enter the elements of the
    array\n");
  for(i=0; i<num; i++)
  {
    scanf("%d", &arr[i]);
  }
  printf("\n Even numbers of the array
      are\n");
  for(i=0; i<num; i++)
  {
    if(arr[i] %2 == 0)
    {
      printf("%d\t", arr[i]);
    }
  }
printf("\n odd numbers of the arr
  are\n");
  for(i=0; i<=num; i++)
  {
    if(arr[i] %2 == 1)
```

```
    {
      printf ("%d \t", arr[i]);
    }
}
getch();
return 0;
}
```

output: Enter size of the array: 8
Enter the elements of the array:
23
34
45
56
67
78
89
90
Even numbers of the array are
34 56 78 90
Odd numbers of the array are
23 45 67 89

(7)
```
#include <stdio.h>
int main()
{
  int arr[100] = {0};
  int i, x, pos, n = 10;
  for (i=0; i<10; i++)
      arr[i] = i+i;
  for (i=0; i<n; i++)
    printf ("%d", arr[i]);
  printf ("\n");
  x = 50;
  pos = 5;
```

```
n++;
for (i = n-1; i >= pos; i--)
 arr[i] = arr[i-1];
arr[pos-1] = x;
for (i=0; i<n; i++)
   printf ("%d", arr[i]);
Printf ("\n");
 return 0;
}
```

Output:

1 2 3 4 5 6 7 8 9 10.

1 2 3 4 50 5 6 2 8 9 10.

(8)

```
#include <stdio.h>
#include <conio.h>
print (int a[], int n)
{
   int i;
   for (i=0; i<n; i++)
   {
    printf ("%d.", a[i]);
   }
}
int main ()
{
   int a[1000], i, n, index, new1;
   printf ("Enter Size of the array : ");
   scanf ("%d", &n);
  printf ("Enter elements in the array:"
   for (i=0; i<n; i++)
{
   scanf ("%d", &a[i]);
}
```

```
printf ("Enter position should b/w greater
        than, %d: ", n);
scanf ("%d", &index);
if (index <= n && index > 0)
{
  printf ("\n, before deletion :");
  print (a, n);
  for (i= index-1 ; i<n-1 ; i++)
  {
   a[i] = a[i+1];
  }
 printf ("\n after deletion :");
  print(a, n-1);
}
else
   printf ("\n invalid input");
return 0;
}
```

output :

Enter size of the array : 5

Enter elements in array : 5

1

20

0

1

Enter position shouldn't greater than 5 : 3

before deletion : 5 1 20 0 1

After deletion : 5 1 0 1

(9)

```
#include <stdio.h>
#include <limits.h>
void print & largest (int arr[], int arr_size
{
   int i, first, second;
```

```
if (arr_size < 2)
{
  printf ("Invalid input");
  return;
}
first = second = INT_MIN;
for (i=0; i< arr_size ; i++)
{
  if (arr[i] > first)
  {
    second = first;
    first = arr[i];
  }
else if (arr[i] > second && arr[i] !=
                                first)
    second = arr[i];
{
if (second == INT_MIN)
    Printf ("There is no second largest
                    element \n");
else
    printf ("The second largest element
          is %d.n", second);
}
int main ()
{
  int arr[] = {12, 35, 1, 10, 34, 1};
  int n = size of (arr) / size of (arr[0]);
  print2 largest (arr, n);
  return 0;
}
```

Output :

The second largest element is 34

(10)
```
#include <stdio.h>
int max (int a, int b)
{
  return ((a>b) ? a : b);
}
int min (int a, int b)
{
  return ((a<b) ? a : b);
}
int median (int arr[]; int size)
{
  if (size % 2 == 0)
    return (arr[size/2] + arr[size/2-1])/2;
  else
    return arr[size/2];
}
int median2SortedArrays (int arr1[],
                         int arr2[], int size)
{
  int med1;
  int med2;
  if (size <= 0) return -1;
  if (size == 1) return (arr1[0] + arr2[0])/2;
  if (size==2) return (max(arr1[0], arr2[0]) +
                       min(arr1[1], arr2[1]))/2;
  med1 = median(arr1, size);
  med2 = median(arr2, size);
  if (med1 == med2) return med1;
  if (med1 < med2)
  {
    return median2SortedArrays (arr2 +
      size/2, arr1, size-size/2);
  }
}
```

```
int main()
{
int i, m, n;
int arr1[] = {1, 5, 13, 24, 8};
int arr2[] = {3, 8, 15, 17, 32};
m = sizeof(arr1) / sizeof(arr1[0]);
n = sizeof(arr2) / sizeof(arr2[0]);
printf("The given array-1 is :");
for(i=0; i<m; i++)
{
printf("%d", arr1[i]);
}
printf("\n");
printf("The given array-2 is :");
for(i=0; i<n; i++)
{
printf("%d", arr2[i]);
}
printf("\n");
printf("\n The median of the 2 Sorted
array is : %d", median2Sorted
Arrays(arr1, arr2, n));
printf("\n");
return 0;
}
```

output :

The given array -1 is : 1 5 13 24 8
The given array -2 is : 3 8 15 17 32
The median of the 2 Sorted array is

(11)

```
#include <stdio.h>
void multiply (int m1, int m2, int mat1[][m2],
               int n1, int n2, int mat2[][n2],
{
    int x, i, j;
    int res[m1][n2];
    for (i=0; i<m1; i++)
    {
        for (j=0; j<n2; j++)
        {
            res[i][j] = 0;
            for (x=0; x<m2; x++)
            {
                *(*(res+i)+j) += *(*mat1
                +i)+x) * *(*(mat2+x)+j);
            }
        }
    }
for (i=0; i<m1; i++)
{
    for (j=0; j<n2; j++)
    {
        printf("%d", *(*(res+i)+j));
    }
    printf("\n");
}
}
int main()
{
    int mat1[][2] = {{2,4}, {3,4}};
    int mat2[][2] = {{1,2}, {1,3}};
    int m1=2, m2=2, n1=2, n2=2;
    multiply(m1, m2, mat1, n1, n2, mat2);
    return 0;
}
```

Output : 6 16
7 18

(12)
```
#include <stdio.h>
int main ()
{
 int a [10][10], transpose [10][10], r, c, i, j;
 printf ("Enter rows and columns :");
 scanf ("%d %d", &r, &c);
 printf ("\n Enter matrix elements : \n");
 for (i = 0; i < r; ++i)
  for (j = 0; j <= c; ++j)
  {
   printf ("enter elements a %d %d : ",
                i+1, j+1);
   scanf ("%d", &a [i][j]);
  }
 printf ("\n Entered matrix : \n");
 for (i = 0; i < r; ++i)
  for (j = 0; j < c; ++j)
  {
   printf ("%d", a[i][j]);
   if (j == c-1)
    printf ("\n");
  }
 for (i = 0; i <= r; ++i)
  for (j = 0; j < c; ++j)
  {
   transpose [j][i] = a[i][j];
  }
 printf ("\n Transpose of the matrix :\n");
 for (i = 0; i < c; ++i)
  for (j = 0; j < r; ++j)
 {
  printf ("%d", transpose [i][j]);
```

```
        if (j == r-1)
            printf("\n");
    }
    return 0;
}
```

Output: Enter rows and columns: 2
3

Enter matrix elements:
Enter element a11: 1
Enter element a12: 4
Enter element a13: 0
Enter element a21: -5
Enter element a22: 2
Enter element a23: 7

Entered matrix:
1 4 0
-5 2 7

Transpose of the matrix:
1 -5
4 2
0 7

(13)
```
#include <stdio.h>
void main()
{
    int i, j, arr[50][50], sum = 0, n, m = 0;
    printf("\n\n Find Sum of left diganals
    of a matrix :\n");
    printf("-------\n");
    printf("Input the size of the square
                matrix :\n");
    scanf("%d", &n);
    m = n;
    printf("Input elements in the first
                matrix :\n");
```

```
for (i=0; i<n; i++)
{
  for (j=0; j<n; j++)
  {
    printf ("element - [%d],[%d]:",i
    scanf ("%d", &arr[i][j]);
  }
}
printf ("The matrix is : \n");
for (i=0; i<n; i++)
{
  for (j=0; j<n; j++)
  printf ("%4d", arr[i][j]);
  printf ("\n");
}
for (i=0; i<n; i++)
{
  m = m-1
  for (j=0; j<n; j++)
  {
    if (j==m)
    {
    sum = sum + arr[i][j];
    }
  }
}
Printf ("Addition of the left diagon
    elements is : %d \n", sum);
}
```

Output:

Sum of left diagonals of the matrix:

Input the Size of the Square matrix: 2

Input elements in the first matrix:

Element - [0], [0]: 1

Element - [0], [1]: 2

Element - [1], [0]: 3

Element - [1], [1]: 4

The matrix is:

1 2

3 4

Addition of the left diagonal elements is: 5

```
#include <stdio.h>
int main (void)
{
  int a [10] [10];
  int i, j, row = 0, col = 0;
  printf ("Enter the order of the matrix
          (m x n): \n");
  printf ("where m: number of rows; and \n");
  printf ("n: number of columns \n");
  scanf ("%d %d", &row, &col);
  int flag = 0;
  printf ("Enter the elements of the
          matrix \n");
  for (i=0; i<row; i++)
  {
    for (j=0; j<col; j++)
    {
      scanf ("%d", &a[i][j]);
    }
  }
```

```
for ( i = 0; i < row; i++ )
{
  for ( j = 0; j < col; j++ )
  {
    if ( i == j && a[i][i] != 1 )
    {
      flag = -1;
      break;
    }
    else if ( i != j && a[i][j] != 0 )
    {
      flag = -1;
      break;
    }
  }
}
if ( flag == 0 )
{
  printf( "%d is a identify matrix\n" );
}
else
  {
  printf ( "%d is not a identify
                matrix" );
  }
  return 0;
}
```

(15)
```
#include <stdio.h>
int search (int mat[4][4], int n, int x)
{
  if (n == 0)
    return -1;
  int smallest = mat[0][0], largest = mat
                 [n-1][n-1];
  if (x < smallest || x > largest)
    return -1;
  int i = 0, j = n-1;
  while (i < n && j >= 0)
  {
    if (mat[i][j] == x)
    {
      printf ("\n found at %d, %d", i, j);
      return 1;
    }
    if (mat[i][j] > x)
      j--;
    else
      i++;
  }
  printf ("\n Element not found");
  return 0;
}
int main()
{
  int mat[4][4] = {
    {10, 20, 30, 40};
    {15, 25, 35, 45};
    {27, 29, 37, 48};
    {32, 33, 39, 50};
  };
  search (mat, 4, 29);
```

```
    return 0;
}
```

output :

```
found at 2, 1
```