

GLA UNIVERSITY, MATHURA



C-PROGRAMMING LANGUAGE **(WEEK QUESTIONS)**

NAME: SHREYA MUDGAL

CLASS: AY-2

CLASS ROLL NO: 56

SUBJECT: COMPUTER LAB FILE

SUBMITTED TO: GURPREET KAUR MAM

WEEK – 5 Questions:

Q-1. Write a program to print the following patterns:

a. *****

***** **Ans.**

```
#include<stdio.h> int
main() { for(int
i=1;i<=4;i++)
{
    for(int j=1;j<=5;j++)
    {
        printf("*");
    }
    printf("\n");
}

return 0;
```

```
}
```

Output:

```
C:\Users\Sarthak\Docun  ×  +  v
*****
*****
*****
*****

-----
Process exited after 0.6244 seconds with return value 0
Press any key to continue . . .
```

b. *

**

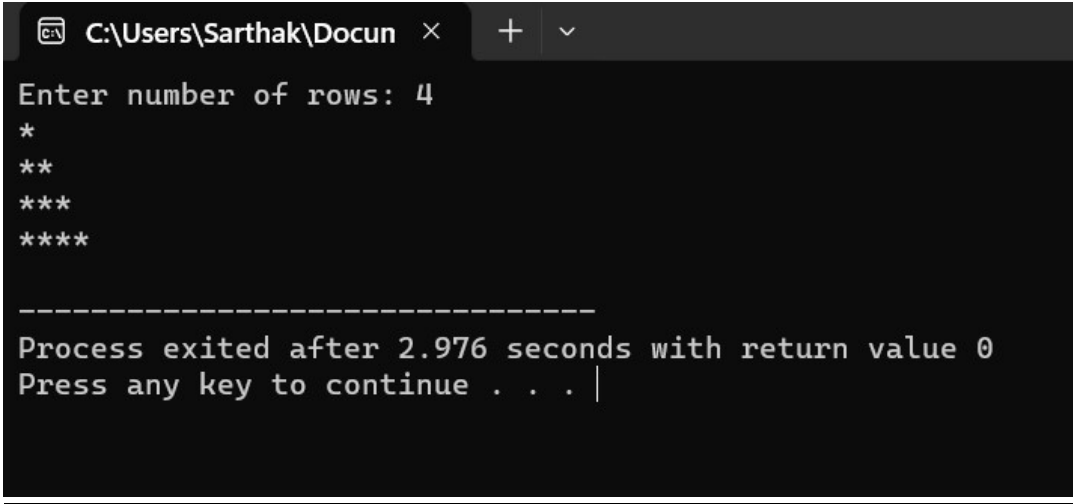
Ans. #include<stdio.h> int

main() {

int n; printf("Enter number of
rows: "); scanf("%d", &n);

```
for(int i=1;i<=n;i++){  
    for(int j=1;j<=i;j++){  
        {  
            printf("*");  
        }  
    }  
    printf("\n");  
}  
return 0;  
}
```

Output:



```
C:\Users\Sarthak\Docun > Enter number of rows: 4  
*  
**  
***  
****  
  
-----  
Process exited after 2.976 seconds with return value 0  
Press any key to continue . . . |
```

c. 54321

5432

543

54

5

Ans.

Output:

```
C:\Users\Sarthak\Docun × + v
Enter number of rows: 4
1
12
123
1234

-----
Process exited after 2.338 seconds with return value 0
Press any key to continue . . . |
```

d.

1

22

333

4444

Ans. #include <stdio.h> int
main()

```
{ int n, i, j;  
printf("Enter the number of rows:  
"); scanf("%d", &n); for(i = 1; i <=  
n; i++)  
{  
for(j = 1; j <= i; j++)  
{  
printf("%d", i);  
}  
printf("\n");  
}  
return 0;  
}
```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter the number of rows: 4
1
22
333
4444

-----
Process exited after 2.811 seconds with return value 0
Press any key to continue . . .
```

e.

54321

5432

543

54

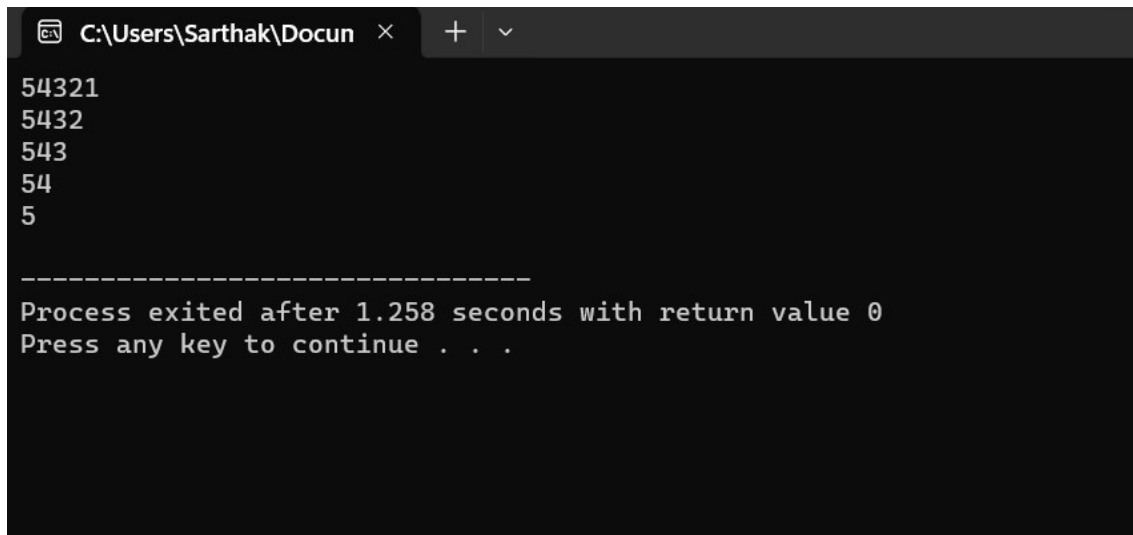
5

Ans.

```
#include <stdio.h> int main() {
int rows = 5;    for (int i = 1; i
<= rows; i++) {    for (int j =
rows; j >= i; j--) {
printf("%d", j);
```

```
    }  
    printf("\n");  
}  
return 0;  
}
```

Output:



```
C:\Users\Sarthak\Docun > 54321  
5432  
543  
54  
5  
  
-----  
Process exited after 1.258 seconds with return value 0  
Press any key to continue . . .
```

f.

A

AB

ABC

ABCD


```

Ans. #include <stdio.h> int
main() {
    int i, j;
    int n;

    printf("Enter the number of rows: ");
    scanf("%d", &n);    for (i = 1; i <= n; i++) {        for
(j = 1; j <= n - i; j++) {            printf(" ");
        }

        for (j = 1; j <= i; j++) {
printf("%c", 'A' + j - 1);
        }

        printf("\n");
    }

    return 0;
}

```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter the number of rows: 4
A
AB
ABC
ABCD

-----
Process exited after 4.714 seconds with return value 0
Press any key to continue . . . |
```

Week-6 Questions:

Q-1. Write a menu driven program to insert and delete elements of kth position to an array of size N.

Ans-1.

```
#include <stdio.h> int
main ()
{ int a; printf("enter the no of the elements of the
array:- "); scanf("%d",&a); int n[a]; for(int
i=0;i<a;i++)
{
printf("enter the %d element of the array:- ",i+1);
scanf("%d",&n[i]);
```

```
} int k; printf("enter the element which u want to  
delete:- "); scanf("%d",&k);  
  
int g; printf("enter the element insert behalf of  
delete element:- "); scanf("%d",&g); for(int  
i=0;i<a;i++)  
{  
if(n[i]==k)  
{  
n[i]=g;  
}  
else  
{  
printf("not found!");  
break;  
}  
}  
for(int i=0;i<a;i++)  
{  
printf("\n%d ",n[i]);  
}  
return 0;  
}
```

Output:

```
C:\Users\Sarthak\Docun x + v
enter the no of the elements of the array:- 3
enter the 1 element of the array:- 24
enter the 2 element of the array:- 76
enter the 3 element of the array:- 98
enter the element which u want to delete:- 76
enter the element insert behalf of delete element:- 24
not found!
24
76
98
-----
Process exited after 26.22 seconds with return value 0
Press any key to continue . . . |
```

Q-2. Write the program to print the biggest and smallest element in an array.

Ans-2.

```
#include <stdio.h>

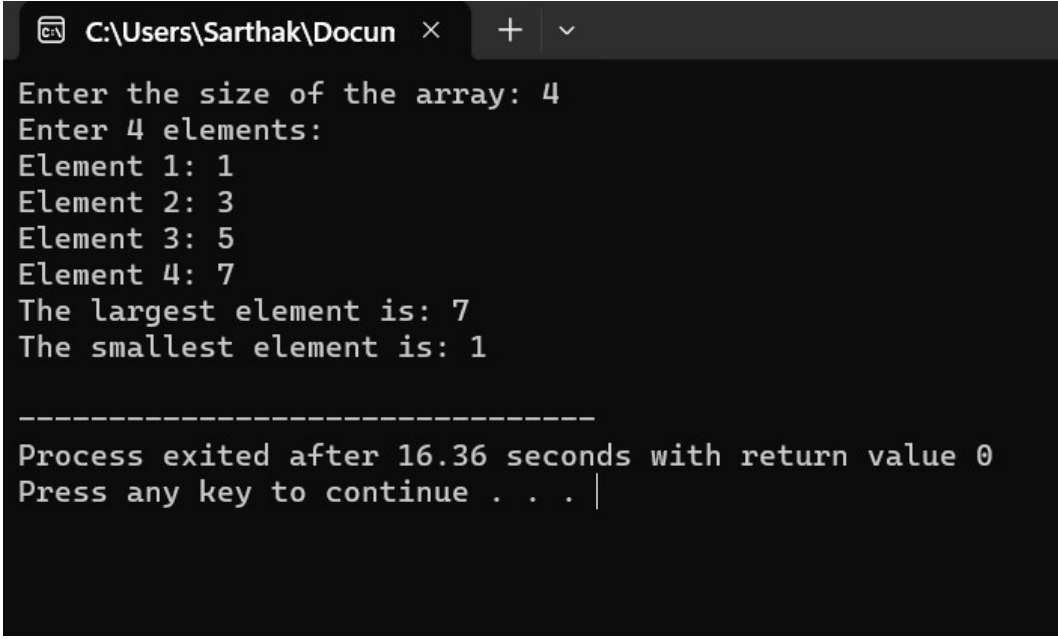
int main() { int
size;

// Get the size of the array
printf("Enter the size of the array: ");
scanf("%d", &size); // Check for
invalid input if (size <= 0) {
printf("Invalid array size.
Exiting...\n"); return 1;
```

```
}  
  
// Create an array of integers  
int arr[size];  
  
// Get elements from the user  
printf("Enter %d elements:\n", size);  
for (int i = 0; i < size; i++) {  
    printf("Element %d: ", i + 1);  
    scanf("%d", &arr[i]);  
}  
  
// Initialize variables for the largest and smallest  
elements  
int largest = arr[0];  
int smallest = arr[0];  
  
// Find the largest and smallest  
elements for (int i = 1; i < size; i++) { if  
(arr[i] > largest) { largest = arr[i];  
}  
if (arr[i] < smallest) { smallest  
= arr[i];  
}  
}
```

```
// Print the results printf("The largest element is:
%d\n", largest); printf("The smallest element is:
%d\n", smallest); return 0;
}
```

Output:

A screenshot of a terminal window with a dark background. The window title bar shows 'C:\Users\Sarthak\Docun' and standard window controls. The terminal displays the following text: 'Enter the size of the array: 4', 'Enter 4 elements:', 'Element 1: 1', 'Element 2: 3', 'Element 3: 5', 'Element 4: 7', 'The largest element is: 7', 'The smallest element is: 1'. Below this, a dashed line separates the output from a status message: 'Process exited after 16.36 seconds with return value 0' and 'Press any key to continue . . . |'.

```
C:\Users\Sarthak\Docun × + v
Enter the size of the array: 4
Enter 4 elements:
Element 1: 1
Element 2: 3
Element 3: 5
Element 4: 7
The largest element is: 7
The smallest element is: 1

-----
Process exited after 16.36 seconds with return value 0
Press any key to continue . . . |
```

Q-3. Write the program to print the sum and average of an array.

Ans-3.

```
#include <stdio.h> int main() { int
size; printf("Enter the size of the
array: "); scanf("%d", &size); if (size
```

```
<= 0) { printf("Invalid array size.  
Exiting...\n"); return 1;  
  
} int arr[size]; printf("Enter %d  
elements:\n", size); for (int i = 0; i <  
size; i++) { printf("Element %d: ", i +  
1); scanf("%d", &arr[i]);  
}  
  
int sum = 0; for (int i = 0; i  
< size; i++) { sum +=  
arr[i];  
}  
  
float average = (float)sum / size; printf("The  
sum of the elements is: %d\n", sum);  
printf("The average of the elements is: %.2f\n",  
average); return 0;  
}
```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter the size of the array: 4
Enter 4 elements:
Element 1: 2
Element 2: 6
Element 3: 3
Element 4: 9
The sum of the elements is: 20
The average of the elements is: 5.00

-----
Process exited after 20.2 seconds with return value 0
Press any key to continue . . . |
```

Q-4. Write the program to sort an array using bubble sort.

Ans-4.

```
#include <stdio.h> void
swap(int *a, int *b) { int
temp = *a; *a = *b;
*b = temp;
}
void bubbleSort(int arr[], int size) {
for (int i = 0; i < size - 1; i++) {
for (int j = 0; j < size - i - 1; j++) {
if (arr[j] > arr[j + 1]) {
swap(&arr[j], &arr[j + 1]);
```



```
}
```

```
}
```

```
}
```

```
}
```

```
int main() { int size; printf("Enter the  
size of the array: "); scanf("%d",  
&size); if (size <= 0) { printf("Invalid  
array size. Exiting...\n"); return 1;  
} int arr[size]; printf("Enter %d  
elements:\n", size); for (int i = 0; i <  
size; i++) { printf("Element %d: ", i +  
1); scanf("%d", &arr[i]);
```

```
}
```

```
bubbleSort(arr, size); printf("\nSorted array  
using Bubble Sort:\n"); for (int i = 0; i < size;  
i++) { printf("%d ", arr[i]);
```

```
}
```

```
printf("\n");
```

```
return 0;
```

```
}
```

Output:

```
C:\Users\Sarthak\Docun x + v
Enter the size of the array: 4
Enter 4 elements:
Element 1: 2
Element 2: 4
Element 3: 6
Element 4: 4

Sorted array using Bubble Sort:
2 4 4 6

-----
Process exited after 26.94 seconds with return value 0
Press any key to continue . . . |
```

Q-5. Write the program to search an element using linear search as well as binary search.

Ans.

```
#include <stdio.h> int linearSearch(int
arr[], int size, int key) { for (int i = 0; i <
size; i++) { if (arr[i] == key) { return i;
}
}
return -1;
}
```

```
int binarySearch(int arr[], int size, int key) {
int low = 0, high = size - 1; while (low <=
high) { int mid = low + (high - low) / 2; if
(arr[mid] == key) { return mid;
} else if (arr[mid] < key) { low
= mid + 1;
} else { high =
mid - 1;
}
}
return -1;
}

int main() { int size, key; printf("Enter
the size of the array: "); scanf("%d",
&size); if (size <= 0) { printf("Invalid
array size. Exiting...\n"); return 1;
}

int arr[size]; printf("Enter %d sorted
elements:\n", size); for (int i = 0; i < size;
i++) { printf("Element %d: ", i + 1);
scanf("%d", &arr[i]);
```

```
}  
  
printf("Enter the element to search: "); scanf("%d",  
&key); int linearIndex = linearSearch(arr, size, key); if  
(linearIndex != -1) { printf("Linear Search: Element  
found at index %d\n", linearIndex);  
} else { printf("Linear Search: Element not  
found\n");  
}  
  
int binaryIndex = binarySearch(arr, size, key); if  
(binaryIndex != -1) { printf("Binary Search: Element  
found at index %d\n", binaryIndex);  
} else { printf("Binary Search: Element not  
found\n");  
}  
  
return 0;  
}
```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter the size of the array: 4
Enter 4 sorted elements:
Element 1: 1
Element 2: 3
Element 3: 5
Element 4: 7
Enter the element to search: -7
Linear Search: Element not found
Binary Search: Element not found

-----
Process exited after 20.84 seconds with return value 0
Press any key to continue . . .
```

Q-6. Take an array of 20 integer inputs from user and print the following:

- a. number of positive numbers**
- b. number of negative numbers**
- c. number of odd numbers**
- d. number of even numbers**
- e. number of 0.**

Ans.

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

```
int main() { int size; printf("Enter the
size of the array: "); scanf("%d",
&size); if (size <= 0) { printf("Invalid
array size. Exiting...\n"); return 1;
}
int arr[size]; printf("Enter %d elements for the
array:\n", size); for (int i = 0; i < size; i++) {
printf("Element %d: ", i + 1); scanf("%d", &(*(arr
+ i))));
}
printf("\nArray elements using pointers:\n");
for (int i = 0; i < size; i++) { printf("%d ", *(arr
+ i));
}
return 0;
}
```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter 20 integer numbers:
Element 1: 1
Element 2: 2
Element 3: 3
Element 4: 4
Element 5: 5
Element 6: 6
Element 7:
7
Element 8: 8
Element 9: 9
Element 10: 0
Element 11: 1
Element 12: 9
Element 13: 8
Element 14: 7
Element 15: 6
Element 16: 5
Element 17: 4
Element 18: 3
Element 19: 2
Element 20: 1

Statistics:
a. Number of positive numbers: 19
b. Number of negative numbers: 0
c. Number of odd numbers: 11
d. Number of even numbers: 9
e. Number of zeros: 1

-----
Process exited after 38.02 seconds with return value 0
Press any key to continue . . . |
```

**Q-7. Take an array of 10 elements.
Split it into middle and store the
elements in two different arrays.**

Ans.

```
#include <stdio.h> int main() { const int size
= 10; int originalArray[size]; int
firstArray[size / 2], secondArray[size / 2];
```

```
printf("Enter %d integer numbers:\n", size);
for (int i = 0; i < size; i++) { printf("Element
%d: ", i + 1); scanf("%d", &originalArray[i]);
} for (int i = 0; i < size / 2; i++) {
firstArray[i] = originalArray[i];
secondArray[i] = originalArray[size / 2 + i];
}
printf("\nOriginal Array:\n"); for
(int i = 0; i < size; i++) {
printf("%d ", originalArray[i]);
}
printf("\n\nSplit Arrays:\n");
printf("First Array:\n");
for (int i = 0; i < size / 2; i++) {
printf("%d ", firstArray[i]);
}
printf("\nSecond Array:\n");
for (int i = 0; i < size / 2; i++) {
printf("%d ", secondArray[i]);
}
return 0;
```



```
}
```

Output:

```
C:\Users\Sarthak\Docun x + v
Enter 10 integer numbers:
Element 1: 3
Element 2: 6
Element 3: -4
Element 4: -36
Element 5: 90
Element 6: 83
Element 7: 12
Element 8: 54
Element 9: 66
Element 10: 99

Original Array:
3 6 -4 -36 90 83 12 54 66 99

Split Arrays:
First Array:
3 6 -4 -36 90
Second Array:
83 12 54 66 99
-----
Process exited after 32.38 seconds with return value 0
Press any key to continue . . . |
```

Q-8. Write the program to count frequency of each element in an array.

Ans.

```
#include <stdio.h> void
toggleChars(char str[])
{ for (int i = 0; str[i] != '\0'; i++) {
if (str[i] >= 'A' && str[i] <= 'Z')
str[i] = str[i] + 'a' - 'A'; else if
```

```
(str[i] >= 'a' && str[i] <= 'z') str[i]
= str[i] + 'A' - 'a';
}
}
// Driver code
int
main()
{
char str[] = "GeKf@rGeek$";
toggleChars(str); printf("String
after toggle \n");
printf("%s\n", str); return 0;
}
```

Output:

```
C:\Users\Sarthak\Docun x + v
Enter 10 integer numbers:
Element 1: 2
Element 2: 4
Element 3: 6
Element 4: 8
Element 5: 3
Element 6: 6
Element 7: 9
Element 8: 1
Element 9: 5
Element 10: 8

Frequency of each element:
2 occurs 1 times
4 occurs 1 times
6 occurs 2 times
8 occurs 2 times
3 occurs 1 times
9 occurs 1 times
1 occurs 1 times
5 occurs 1 times

-----
Process exited after 27.02 seconds with return value 0
Press any key to continue . . . |
```

Week-7 Questions:

Q-1. Write the program to print row major and column major matrix.

Ans. #include <stdio.h>

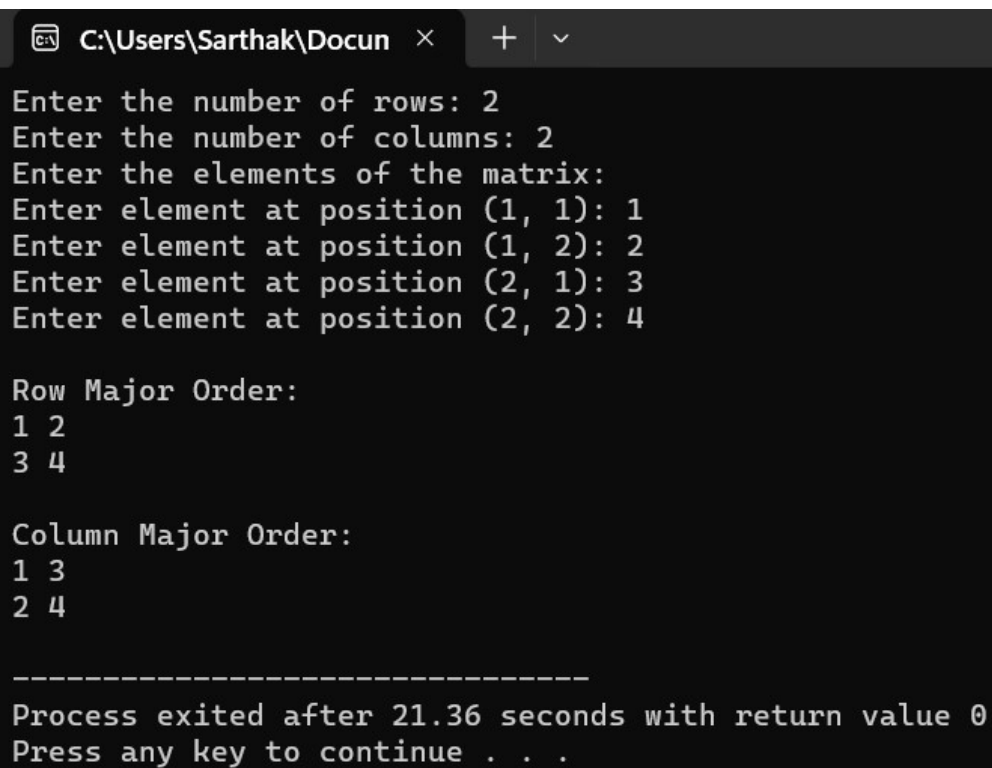
```

int main() {   int rows, cols;   printf("Enter the
number of rows: ");   scanf("%d", &rows);
printf("Enter the number of columns: ");
scanf("%d", &cols);   int matrix[rows][cols];
printf("Enter the elements of the matrix:\n");   for
(int i = 0; i < rows; i++) {       for (int j = 0; j < cols; j++)
{           printf("Enter element at position (%d, %d): ",
i +
1, j + 1);           scanf("%d",
&matrix[i][j]);
        }
    }
    printf("\nRow Major Order:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
    printf("\nColumn Major Order:\n");

```

```
        for (int j = 0; j < cols; j++) {  
for (int i = 0; i < rows; i++) {  
printf("%d ", matrix[i][j]);  
        }  
        printf("\n");  
    }  
    return 0;  
}
```

Output:



```
C:\Users\Sarthak\Docun > Enter the number of rows: 2  
Enter the number of columns: 2  
Enter the elements of the matrix:  
Enter element at position (1, 1): 1  
Enter element at position (1, 2): 2  
Enter element at position (2, 1): 3  
Enter element at position (2, 2): 4  
  
Row Major Order:  
1 2  
3 4  
  
Column Major Order:  
1 3  
2 4  
  
-----  
Process exited after 21.36 seconds with return value 0  
Press any key to continue . . .
```

Q-2. Write the program to print sum of a whole matrix:

Ans. #include <stdio.h> int main() { int rows, cols; printf("Enter the number of rows: "); scanf("%d", &rows); printf("Enter the number of columns: "); scanf("%d", &cols); int matrix[rows][cols]; printf("Enter the elements of the matrix:\n"); for (int i = 0; i < rows; i++) { for (int j = 0; j < cols; j++) { printf("Enter element at position (%d, %d): ", i + 1, j + 1); scanf("%d", &matrix[i][j]); } } printf("\nRow Major Order:\n"); for (int i = 0; i < rows; i++) { for (int j = 0; j < cols; j++) { printf("%d ", matrix[i][j]); } printf("\n"); }

```

    }

    printf("\nColumn Major Order:\n");

    for (int j = 0; j < cols; j++) {
    for (int i = 0; i < rows; i++) {
    printf("%d ", matrix[i][j]);

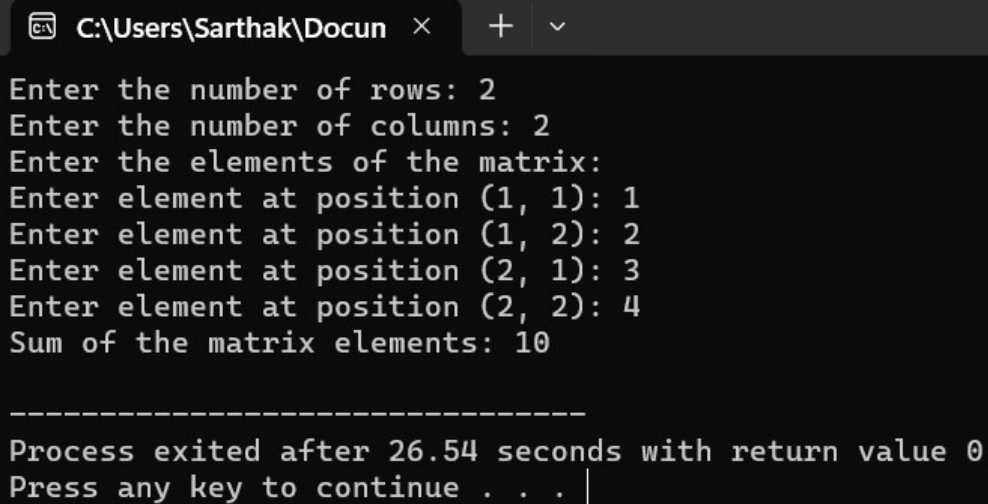
        }

        printf("\n");
    }

    return 0;
}

```

Output:



```

C:\Users\Sarthak\Docun >
Enter the number of rows: 2
Enter the number of columns: 2
Enter the elements of the matrix:
Enter element at position (1, 1): 1
Enter element at position (1, 2): 2
Enter element at position (2, 1): 3
Enter element at position (2, 2): 4
Sum of the matrix elements: 10

-----
Process exited after 26.54 seconds with return value 0
Press any key to continue . . . |

```

Q-3. Write a program to add and multiply two 3x3 matrices. You can use 2D array to create a matrix.

Ans. #include <stdio.h>

```
int main() {  
    int rows, cols; printf("Enter the number of rows for  
the matrices: ");  
    scanf("%d", &rows);    printf("Enter the number of  
columns for the matrices: ");    scanf("%d", &cols);  
    int matrix1[rows][cols], matrix2[rows][cols],  
    sumMatrix[rows][cols], productMatrix[rows][cols];  
    printf("\nEnter elements for the first matrix:\n");  
    for (int i = 0; i < rows; i++) {        for (int j = 0; j < cols;  
j++) {            printf("Enter element at position (%d,  
%d): ", i +  
1, j + 1);            scanf("%d",  
&matrix1[i][j]);  
        }  
    }  
  
    printf("\nEnter elements for the second matrix:\n");  
    for (int i = 0; i < rows; i++) {        for (int j = 0; j < cols;
```



```

j++) {      printf("Enter element at position (%d,
%d): ", i + 1, j + 1);

        scanf("%d", &matrix2[i][j]);

    }

}

for (int i = 0; i < rows; i++) {

}

for (int i = 0; i < rows; i++) {      for (int j =
0; j < cols; j++) {      productMatrix[i][j] =
0;      for (int k = 0; k < cols; k++) {
productMatrix[i][j] += matrix1[i][k] *
matrix2[k][j];

        }

    }

}

printf("\nSum of the matrices:\n");

for (int i = 0; i < rows; i++) {      for
(int j = 0; j < cols; j++) {
printf("%d ", sumMatrix[i][j]);

    }

    printf("\n");

}

```

```
    printf("\nProduct of the matrices:\n");  
    for (int i = 0; i < rows; i++) {        for (int j  
= 0; j < cols; j++) {            printf("%d ",  
productMatrix[i][j]);  
        }  
        printf("\n");  
  
        for (int j = 0; j < cols; j++) {  
sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];  
        }  
    }  
    return 0;  
}
```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter the number of rows for the matrices: 2
Enter the number of columns for the matrices: 2

Enter elements for the first matrix:
Enter element at position (1, 1): 1
Enter element at position (1, 2): 4
Enter element at position (2, 1): 8
Enter element at position (2, 2): 4

Enter elements for the second matrix:
Enter element at position (1, 1): 2
Enter element at position (1, 2): 3
Enter element at position (2, 1): 6
Enter element at position (2, 2): 7

Sum of the matrices:
3 7
14 11

Product of the matrices:
26 31
40 52

-----
Process exited after 26.73 seconds with return value 0
Press any key to continue . . . |
```

Q-4. Write the program to print sum of all diagonal elements, upper triangular matrix and lower triangular matrix.

Ans.

```
#include <stdio.h>

int main() { int size; printf("Enter the size of the
square matrix: "); scanf("%d", &size); int
matrix[size][size]; printf("Enter the elements of the
```

```

matrix:\n");    for (int i = 0; i < size; i++) {        for (int j
= 0; j < size; j++) {            printf("Enter element at
position (%d, %d): ", i +
1, j + 1);        scanf("%d",
&matrix[i][j]);
        }
    }

    int sumDiagonal = 0;    for
(int i = 0; i < size; i++) {
sumDiagonal += matrix[i][i];
    }

    int sumUpperTriangular = 0;
for (int i = 0; i < size; i++) {
for (int j = i + 1; j < size; j++) {
sumUpperTriangular +=
matrix[i][j];
        }
    }

    int sumLowerTriangular = 0;    for (int i
= 0; i < size; i++) {        for (int j = 0; j < i;

```

```
j++) {      sumLowerTriangular +=
matrix[i][j];
    }
}

printf("\nSum of diagonal elements: %d\n",
sumDiagonal);  printf("Sum of upper triangular
elements: %d\n", sumUpperTriangular);
printf("Sum of lower triangular elements: %d\n",
sumLowerTriangular);

return 0;
}
```

Output:

Q-5. Write the program to find the

```
C:\Users\Sarthak\Docun × + v
Enter the size of the square matrix: 4
Enter the elements of the matrix:
Enter element at position (1, 1): 1
Enter element at position (1, 2): 4
Enter element at position (1, 3): 7
Enter element at position (1, 4): 0
Enter element at position (2, 1): 3
Enter element at position (2, 2): 0
Enter element at position (2, 3): 3
Enter element at position (2, 4): 5
Enter element at position (3, 1): 5
Enter element at position (3, 2): 8
Enter element at position (3, 3): 2
Enter element at position (3, 4): 1
Enter element at position (4, 1): 5
Enter element at position (4, 2): 9
Enter element at position (4, 3): 0
Enter element at position (4, 4): 5

Sum of diagonal elements: 8
Sum of upper triangular elements: 20
Sum of lower triangular elements: 30

-----
Process exited after 680.1 seconds with return value 0
Press any key to continue . . . |
```

**frequency of odd and even
elements in matrix.\ Ans.**

```
#include <stdio.h>
int main() {    int rows, cols;

printf("Enter the number of rows: ");
scanf("%d", &rows);    printf("Enter the
```

```

number of columns: ");    scanf("%d",
&cols);    int matrix[rows][cols];
printf("Enter the matrix elements:\n");
for (int i = 0; i < rows; i++) {    for (int j =
0; j < cols; j++) {        scanf("%d",
&matrix[i][j]);
    }
}

int oddFrequency = 0, evenFrequency =
0;    for (int i = 0; i < rows; i++) {    for (int
j = 0; j < cols; j++) {        if (matrix[i][j] % 2
== 0) {            evenFrequency++;
        } else {
            oddFrequency++;
        }
    }
}

printf("Frequency of odd elements: %d\n",
oddFrequency);    printf("Frequency of even
elements: %d\n", evenFrequency);

return 0;

}

```

Output:

```
C:\Users\Sarthak\Docun x + v
Enter the number of rows: 2
Enter the number of columns: 2
Enter the matrix elements:
1
2
3
4
Frequency of odd elements: 2
Frequency of even elements: 2

-----
Process exited after 17.03 seconds with return value 0
Press any key to continue . . . |
```

Q-6. Write the program to find sum of each row and sum of each column of matrix.

Ans.

```
#include <stdio.h> int main() {    int rows,
cols;    printf("Enter the number of rows:
");    scanf("%d", &rows);    printf("Enter
the number of columns: ");    scanf("%d",
&cols);    int matrix[rows][cols];
printf("Enter the matrix elements:\n");
for (int i = 0; i < rows; i++) {        for (int j =
```



```

0; j < cols; j++) {          scanf("%d",
&matrix[i][j]);
    }
}

for (int i = 0; i < rows; i++) {
    int rowSum = 0;    for
(int j = 0; j < cols; j++) {
rowSum += matrix[i][j];
    }
    printf("Sum of row %d: %d\n", i + 1, rowSum);
}

for (int j = 0; j < cols; j++) {
int colSum = 0;    for (int i =
0; i < rows; i++) {          colSum
+= matrix[i][j];
    }
    printf("Sum of column %d: %d\n", j + 1, colSum);
}

return 0;
}

```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter the number of rows: 3
Enter the number of columns: 3
Enter the matrix elements:
1
2
3
4
5
6
7
8
9
Sum of row 1: 6
Sum of row 2: 15
Sum of row 3: 24
Sum of column 1: 12
Sum of column 2: 15
Sum of column 3: 18

-----
Process exited after 24.85 seconds with return value 0
Press any key to continue . . . |
```

Q-7. Initialize a 2D array of 3*3 matrix.

Ans.

```
#include <stdio.h> int main() {   int matrix[3][3];
printf("Enter the elements for a 3x3 matrix:\n");
for (int i = 0; i < 3; i++) {      for (int j = 0; j < 3; j++)
{
    printf("Enter element at position [%d][%d]: ", i +
1, j + 1);      scanf("%d",
&matrix[i][j]);
```

```

    }
}

printf("\nMatrix in a square box:\n");
printf("+---+---+---+\n"); for (int i = 0; i < 3;
i++) { printf("| %d | %d | %d |\n",
matrix[i][0], matrix[i][1], matrix[i][2]);
    if (i < 2) { printf("+---+---+---+
-+\n");
    }
}

printf("+---+---+---+\n");
return 0;
}

```

Output:

Q-8. A square matrix, one having the

```
C:\Users\Sarthak\Docun × + v
Enter the elements for a 3x3 matrix:
Enter element at position [1][1]: 3
Enter element at position [1][2]: 6
Enter element at position [1][3]: 9
Enter element at position [2][1]: 2
Enter element at position [2][2]: 4
Enter element at position [2][3]: 6
Enter element at position [3][1]: 3
Enter element at position [3][2]: 9
Enter element at position [3][3]: 12

Matrix in a square box:
+---+---+---+
| 3 | 6 | 9 |
+---+---+---+
| 2 | 4 | 6 |
+---+---+---+
| 3 | 9 | 12 |
+---+---+---+

-----
Process exited after 27.63 seconds with return value 0
Press any key to continue . . . |
```

same number of rows and columns, is called a diagonal matrix if it's only non-zero elements are on the diagonal from upper left to lower right. It is called upper triangular matrix if all elements bellow the diagonal are zeroes, and lower triangular matrix, if

all the elements above the diagonal are zeroes. Write a program that reads a matrix and determines if it is one of the above mentioned three special matrices.

Ans.

```
#include <stdio.h> int main() {    int n;
printf("Enter the size of the square matrix: ");
scanf("%d", &n);    int matrix[n][n];
printf("Enter the matrix elements:\n");    for
(int i = 0; i < n; i++) {        for (int j = 0; j < n; j++)
{            scanf("%d", &matrix[i][j]);
        }
    }

    int isDiagonal = 1;    int
isUpperTriangular = 1;    int
isLowerTriangular = 1;    for (int i =
0; i < n; i++) {        for (int j = 0; j <
n; j++) {            if (i != j &&
```

```

matrix[i][j] != 0) {
isDiagonal = 0;
    }
    if (i < j && matrix[i][j] != 0) {
isUpperTriangular = 0;
    }
    if (i > j && matrix[i][j] != 0) {
isLowerTriangular = 0;
    }
    }
}

if (isDiagonal) {    printf("The matrix is a
diagonal matrix.\n");
    } else if (isUpperTriangular) {
    printf("The matrix is an upper triangular matrix.\n");
    } else if (isLowerTriangular) {    printf("The matrix
is a lower triangular matrix.\n");
    } else {    printf("The matrix is not
diagonal, upper triangular, or lower
triangular.\n");
    }

return 0;

```

}

Output:

```
C:\Users\Sarthak\Docun x + v
Enter the elements for a 3x3 matrix:
Enter element at position [1][1]: 3
Enter element at position [1][2]: 6
Enter element at position [1][3]: 9
Enter element at position [2][1]: 2
Enter element at position [2][2]: 4
Enter element at position [2][3]: 6
Enter element at position [3][1]: 3
Enter element at position [3][2]: 9
Enter element at position [3][3]: 12

Matrix in a square box:
+---+---+---+
| 3 | 6 | 9 |
+---+---+---+
| 2 | 4 | 6 |
+---+---+---+
| 3 | 9 | 12 |
+---+---+---+

-----
Process exited after 27.63 seconds with return value 0
Press any key to continue . . . |
```

Q-9. Write the program to check whether the matrix is sparse matrix or not. Ans.

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

```
int main() {    int
```

```
rows, cols;
```

```
printf("Enter the
```

```
number of rows:
");  scanf("%d",
&rows);
printf("Enter the
number of
columns: ");
scanf("%d",
&cols);  int
matrix[rows][cols]
;  printf("Enter
the matrix
elements:\n");
for (int i = 0; i <
rows; i++) {
for (int j = 0; j <
cols; j++) {
scanf("%d",
&matrix[i][j]);
    }
}
```



```
    int nonZeroCount = 0;    for
(int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
    if (matrix[i][j] != 0) {
    nonZeroCount++;
        }
    }
    }
    int threshold = rows * cols / 2;
    if (nonZeroCount <= threshold) {
    printf("The matrix is a sparse matrix.\n");
    } else {    printf("The matrix is not a sparse
matrix.\n");
    }
    return 0;
}
```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter the number of rows: 2
Enter the number of columns: 2
Enter the matrix elements:
2
2
0
0
The matrix is a sparse matrix.

-----
Process exited after 20.17 seconds with return value 0
Press any key to continue . . . |
```

Week-8 Questions:

Q-1. Write a C program to create, initialize and use pointers.

Ans.

```
#include <stdio.h> int main() { int number =
42; float floatNumber = 3.14; char character
= 'A'; int *intPointer; float *floatPointer; char
*charPointer; intPointer = &number;
floatPointer = &floatNumber; charPointer =
&character; printf("Original values:\n");
```

```
printf("Number: %d\n", number);
printf("Float Number: %.2f\n",
floatNumber); printf("Character: %c\n\n",
character); *intPointer = 100;

*floatPointer = 2.718; *charPointer = 'B';
printf("Modified values using pointers:\n");
printf("Number: %d\n", number); printf("Float
Number: %.2f\n", floatNumber);
printf("Character: %c\n\n", character); int
anotherNumber = 10; int *resultPointer =
&number; *resultPointer += anotherNumber;
printf("Result of adding %d to the original
number using pointers: %d\n", anotherNumber,
*resultPointer);
return 0;
}
```

Output:

```
C:\Users\Sarthak\Docun x + v
Original values:
Number: 42
Float Number: 3.14
Character: A

Modified values using pointers:
Number: 100
Float Number: 2.72
Character: B

Result of adding 10 to the original number using pointers: 110

-----
Process exited after 1.026 seconds with return value 0
Press any key to continue . . .
```

Q-2. Write a C program to add two numbers using pointers.

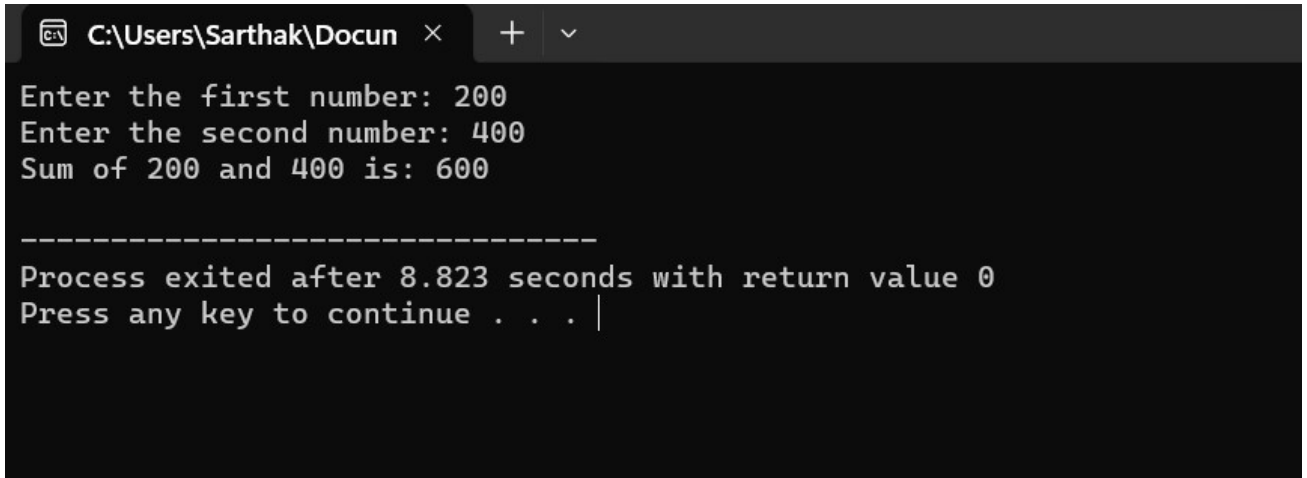
Ans.

```
#include<stdio.h>

int main()
{
    int num1, num2, sum; int *ptr1,
    *ptr2; ptr1 = &num1; ptr2 =
    &num2; printf("Enter the first
    number: "); scanf("%d", ptr1);
    printf("Enter the second
    number: "); scanf("%d", ptr2);
    sum = *ptr1 + *ptr2;
```

```
printf("Sum of %d and %d is:
%d\n", *ptr1, *ptr2, sum);
return 0;
}
```

Output:



```
C:\Users\Sarthak\Docun >
Enter the first number: 200
Enter the second number: 400
Sum of 200 and 400 is: 600

-----
Process exited after 8.823 seconds with return value 0
Press any key to continue . . . |
```

Q-3. Write a C program to swap two numbers using pointers.

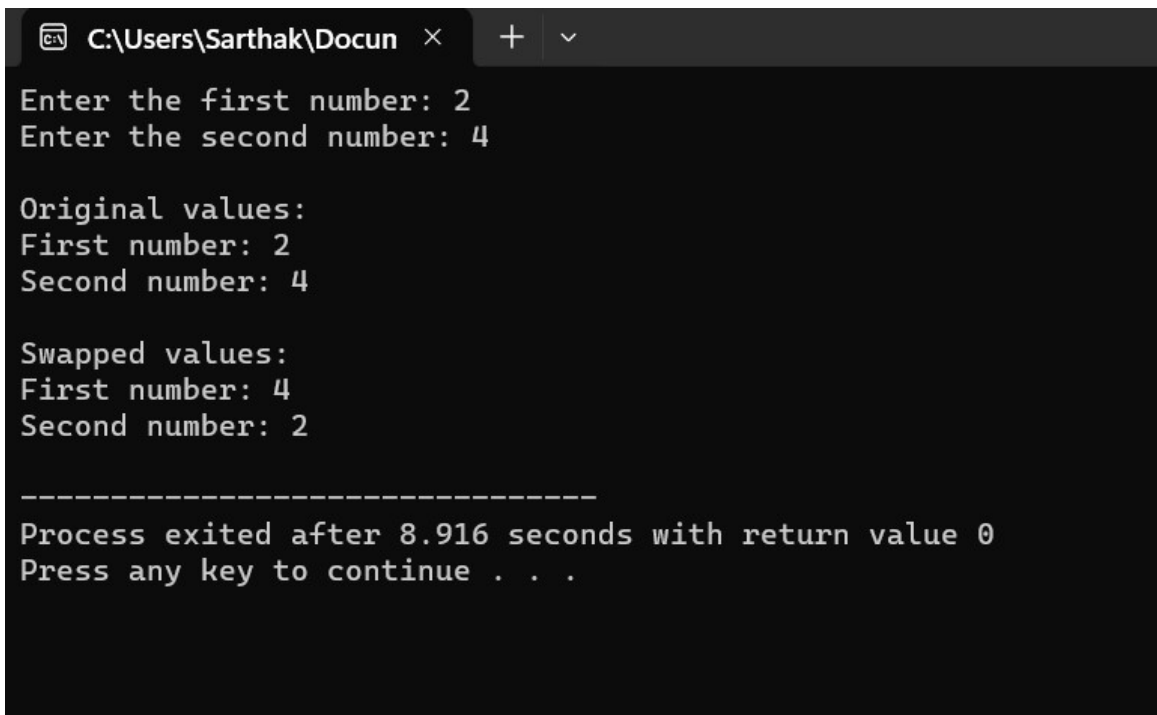
Ans.

```
#include <stdio.h> void swap(int
*num1, int *num2) { int temp =
*num1; *num1 = *num2;
*num2 = temp;
}

int main() { int num1, num2;
printf("Enter the first number: ");
```

```
scanf("%d", &num1); printf("Enter the  
second number: "); scanf("%d",  
&num2); printf("\nOriginal values:\n");  
printf("First number: %d\n", num1);  
printf("Second number: %d\n", num2);  
swap(&num1, &num2);  
printf("\nSwapped values:\n");  
printf("First number: %d\n", num1);  
printf("Second number: %d\n", num2);  
return 0;  
}
```

Output:



```
C:\Users\Sarthak\Docum x + v  
Enter the first number: 2  
Enter the second number: 4  
  
Original values:  
First number: 2  
Second number: 4  
  
Swapped values:  
First number: 4  
Second number: 2  
  
-----  
Process exited after 8.916 seconds with return value 0  
Press any key to continue . . .
```

Q-4. Write a C program to input and print array elements using pointer.

Ans.

```
#include <stdio.h> int main() { int size;
printf("Enter the size of the array: ");
scanf("%d", &size); if (size <= 0) {
printf("Invalid array size. Exiting...\n");
return 1;

}

int arr[size]; printf("Enter %d elements for the
array:\n", size); for (int i = 0; i < size; i++) {
printf("Element %d: ", i + 1); scanf("%d", &*(arr
+ i)));
}

printf("\nArray elements using pointers:\n");
for (int i = 0; i < size; i++) { printf("%d ", *(arr
+ i));
}

return 0; }
```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter the size of the array: 6
Enter 6 elements for the array:
Element 1: 2
Element 2: 4
Element 3: 6
Element 4: 8
Element 5: 0
Element 6: 2

Array elements using pointers:
2 4 6 8 0 2
-----
Process exited after 21.42 seconds with return value 0
Press any key to continue . . .
```

Q-5. Write a C program to copy one array to another using pointer.

Ans.

```
#include<stdio.h> void copyArray(int *source, int
*destination, int size)
{
for (int i = 0; i < size; i++)
{
*(destination + i) = *(source + i);
}
}
int main()
```



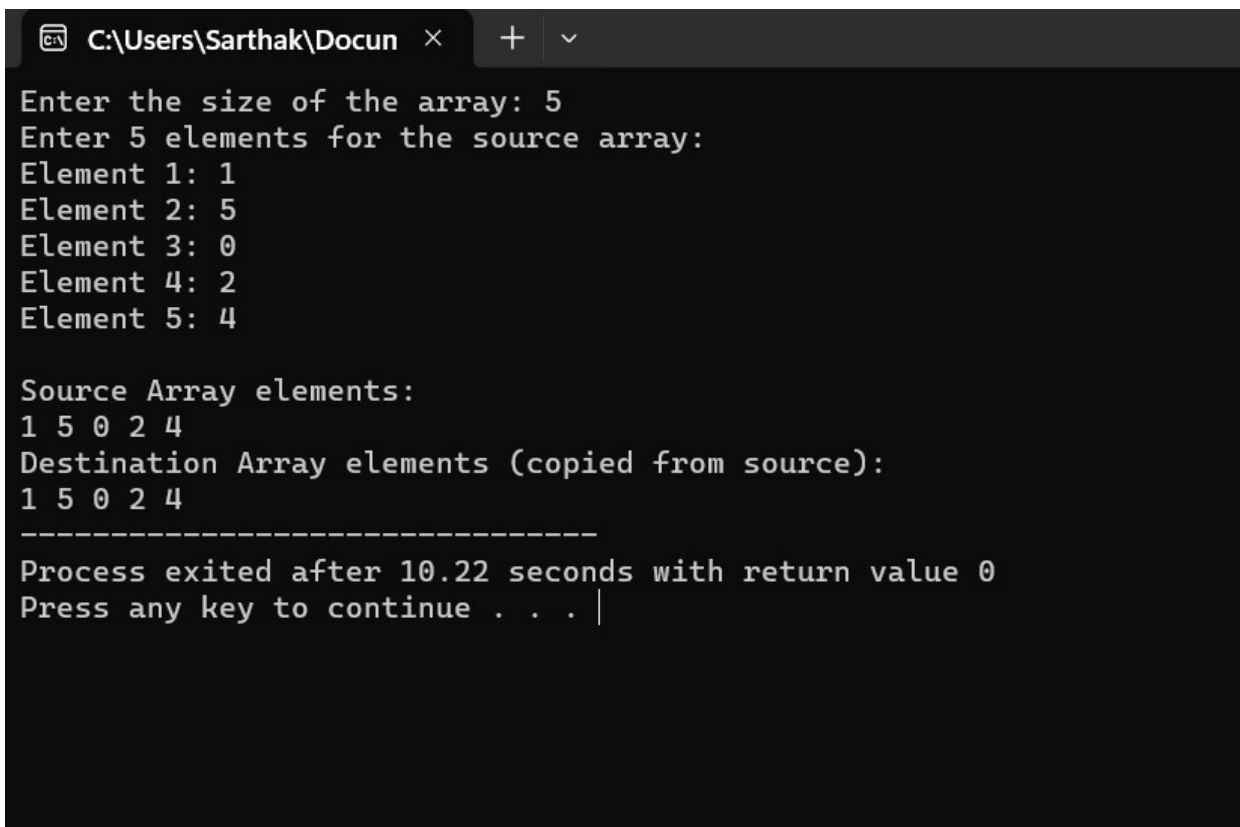
```
{ int size; printf("Enter the size of
the array: "); scanf("%d", &size); if
(size <= 0)
{
printf("Invalid array size. Exiting...\n");
return 1;
}

int sourceArray[size]; int destinationArray[size];
printf("Enter %d elements for the source
array:\n", size); for (int i = 0; i < size; i++)
{
printf("Element %d: ", i + 1);
scanf("%d", &*(sourceArray + i)));
}

copyArray(sourceArray, destinationArray,
size); printf("\nSource Array elements:\n");
for (int i = 0; i < size; i++)
{
printf("%d ", *(sourceArray + i));
}
}
```

```
// Print destination array elements (copied array)
printf("\nDestination Array elements (copied
from source):\n"); for (int i = 0; i < size; i++)
{
printf("%d ", *(destinationArray + i));
}
return 0;
}
```

Output:

A screenshot of a Windows terminal window with a dark background. The window title bar shows the file path 'C:\Users\Sarthak\Docun' and standard window controls. The terminal displays the output of a C program. It starts with prompts for array size and elements, followed by the source array elements, then the destination array elements (copied from source), and finally a message indicating the process exited after 10.22 seconds with a return value of 0, followed by a 'Press any key to continue' prompt.

```
C:\Users\Sarthak\Docun × + v
Enter the size of the array: 5
Enter 5 elements for the source array:
Element 1: 1
Element 2: 5
Element 3: 0
Element 4: 2
Element 5: 4

Source Array elements:
1 5 0 2 4
Destination Array elements (copied from source):
1 5 0 2 4
-----
Process exited after 10.22 seconds with return value 0
Press any key to continue . . . |
```

Q-6. Write a C program to swap two arrays using pointers.

Ans.

```
#include <stdio.h> void swapArrays(int *arr1,
int *arr2, int size) { for (int i = 0; i < size; i++) {
int temp = *(arr1 + i);
*(arr1 + i) = *(arr2 + i);
*(arr2 + i) = temp;
}
}

void printArray(int *arr, int size) {
for (int i = 0; i < size; i++) {
printf("%d ", *(arr + i));
}
printf("\n");
}

int main() { int size; printf("Enter the
size of the arrays: "); scanf("%d",
&size); if (size <= 0) { printf("Invalid
array size. Exiting...\n"); return 1;
}
```

```
int array1[size], array2[size]; printf("Enter %d
elements for the first array:\n", size); for (int i = 0; i <
size; i++) { printf("Element %d: ", i + 1);
scanf("%d", &(*(array1 + i)));
}
printf("\nEnter %d elements for the second
array:\n", size); for (int i = 0; i < size; i++) {
printf("Element %d: ", i + 1); scanf("%d", &(*(array2
+ i)));
}
printf("\nOriginal Arrays:\n");
printf("Array 1: ");
printArray(array1, size);
printf("Array 2: ");
printArray(array2, size);
swapArrays(array1, array2,
size); printf("\nSwapped
Arrays:\n"); printf("Array 1: ");
printArray(array1, size);
printf("Array 2: ");
printArray(array2, size); return
0;
```

}

Output:

```
C:\Users\Sarthak\Docum x + v
Enter the size of the arrays: 8
Enter 8 elements for the first array:
Element 1: 2
Element 2: 3
Element 3: 4
Element 4: 5
Element 5: 6
Element 6: 7
Element 7: 8
Element 8: 9

Enter 8 elements for the second array:
Element 1: 9
Element 2: 8
Element 3: 7
Element 4: 6
Element 5: 5
Element 6: 4
Element 7: 3
Element 8: 2

Original Arrays:
Array 1: 2 3 4 5 6 7 8 9
Array 2: 9 8 7 6 5 4 3 2

Swapped Arrays:
Array 1: 9 8 7 6 5 4 3 2
Array 2: 2 3 4 5 6 7 8 9

-----
Process exited after 33.63 seconds with return value 0
Press any key to continue . . . |
```

Q-7. Write a C program to reverse an array using pointers.

Ans.

```
#include <stdio.h> void
reverseArray(int *arr, int size) { int
*start = arr; int *end = arr + size -
1; while (start < end) { int temp =
*start; *start = *end; *end = temp;
start++; end--;
}
}

void printArray(int *arr, int size) {
for (int i = 0; i < size; i++) {
printf("%d ", *(arr + i));
}
printf("\n");
}

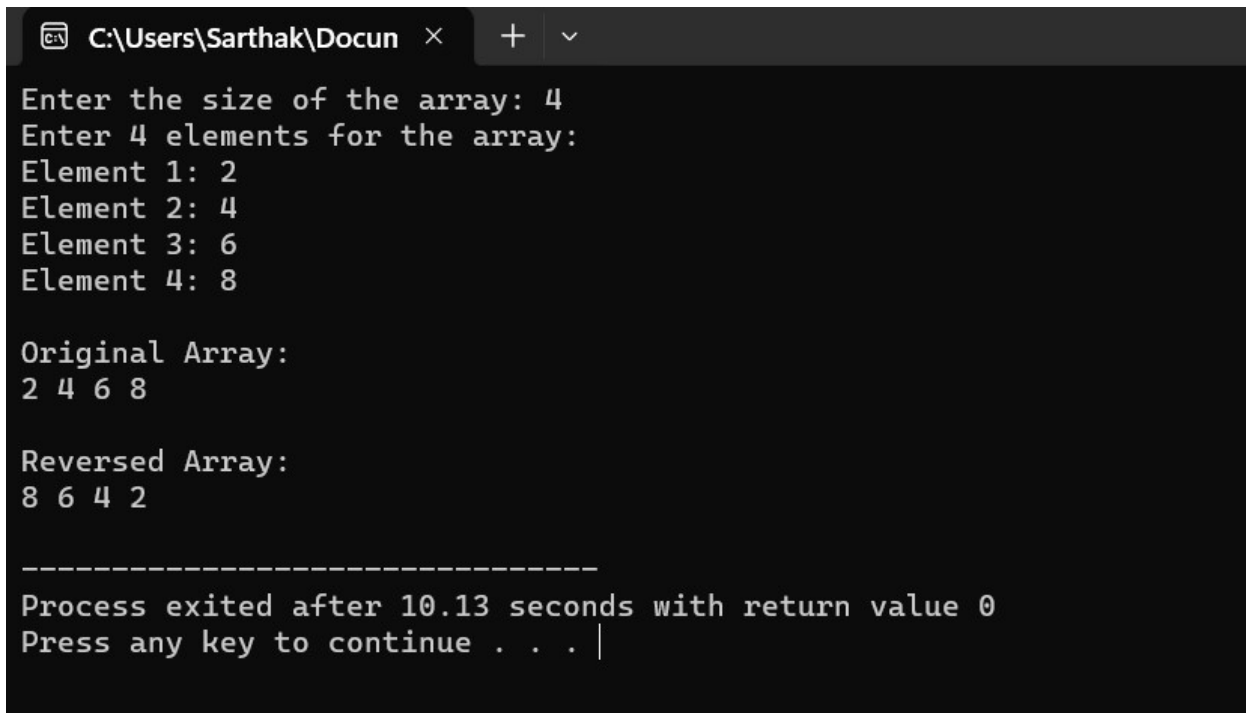
int main() {
int size; printf("Enter the size of the
array: "); scanf("%d", &size); if (size
<= 0) { printf("Invalid array size.
Exiting...\n"); return 1;
}
```

```

int array[size]; printf("Enter %d elements for the
array:\n", size); for (int i = 0; i < size; i++) {
printf("Element %d: ", i + 1); scanf("%d",
&(*(array + i))); printf("\nOriginal Array:\n");
printArray(array, size); reverseArray(array, size);
printf("\nReversed Array:\n"); printArray(array,
size); return 0;
}

```

Output:



```

C:\Users\Sarthak\Docun >
Enter the size of the array: 4
Enter 4 elements for the array:
Element 1: 2
Element 2: 4
Element 3: 6
Element 4: 8

Original Array:
2 4 6 8

Reversed Array:
8 6 4 2

-----
Process exited after 10.13 seconds with return value 0
Press any key to continue . . . |

```

Q-8. Write a C program to add two matrix using pointers.

Ans.

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

```
#define MAX_SIZE 10 void addMatrices(int *matrix1,  
int *matrix2, int *result, int rows, int cols) { for (int i =  
0; i < rows; i++) { for (int j = 0; j < cols; j++) {
```

```
*(result + i * cols + j) = *(matrix1 + i * cols + j) +  
*(matrix2 + i * cols + j);
```

```
}
```

```
}
```

```
}
```

```
void printMatrix(int *matrix, int rows, int cols)
```

```
{ for (int i = 0; i < rows; i++) { for (int j = 0; j <  
cols; j++) { printf("%d ", *(matrix + i * cols + j));
```

```
}
```

```
printf("\n");
```

```
}
```

```
}
```

```
int main() { int rows, cols; printf("Enter  
the number of rows: "); scanf("%d",  
&rows); printf("Enter the number of  
columns: "); scanf("%d", &cols);
```



```

if (rows <= 0 || cols <= 0 || rows > MAX_SIZE || cols
> MAX_SIZE) { printf("Invalid matrix size.
Exiting...\n");
return 1;
}

int matrix1[MAX_SIZE][MAX_SIZE],
matrix2[MAX_SIZE][MAX_SIZE],
result[MAX_SIZE][MAX_SIZE]; printf("Enter elements
for the first matrix (%dx%d):\n", rows, cols); for (int i =
0; i < rows; i++) { for (int j = 0; j < cols; j++) {
printf("Element at (%d, %d): ", i + 1, j + 1); scanf("%d",
&(*(matrix1 + i * cols + j)));
}
}

printf("\nEnter elements for the second matrix
(%dx%d):\n", rows, cols); for (int i = 0; i < rows;
i++) { for (int j = 0; j < cols; j++) {
printf("Element at (%d, %d): ", i + 1, j + 1);
scanf("%d", &(*(matrix2 + i * cols + j))); //
Using pointer notation to access matrix
}
}

```

```
addMatrices((int *)matrix1, (int *)matrix2, (int
*)result, rows, cols); printf("\nOriginal Matrices:\n");
printf("Matrix 1:\n"); printMatrix((int *)matrix1, rows,
cols); printf("Matrix 2:\n"); printMatrix((int *)matrix2,
rows, cols);
printf("\nResult Matrix (Sum of Matrix 1 and Matrix
2):\n"); printMatrix((int *)result,
rows, cols); return 0;
}
```

Output:

Q-9. Write a C program to multiply two matrix using pointers.

Ans.

```
C:\Users\Sarthak\Docun × + v
Enter the number of rows: 2
Enter the number of columns: 2
Enter elements for the first matrix (2x2):
Element at (1, 1): 1
Element at (1, 2): 3
Element at (2, 1): 0
Element at (2, 2): 0

Enter elements for the second matrix (2x2):
Element at (1, 1): 2
Element at (1, 2): 0
Element at (2, 1): 4
Element at (2, 2): 1

Original Matrices:
Matrix 1:
1 0
8257536 0
Matrix 2:
2 0
7541712 0

Result Matrix (Sum of Matrix 1 and Matrix 2):
3 0
15799248 0

-----
Process exited after 22.61 seconds with return value 0
Press any key to continue . . .
```

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

```
#define MAX_SIZE 10
```

```
void multiplyMatrices(int
firstMatrix[MAX_SIZE][MAX_SIZE], int
secondMatrix[MAX_SIZE][MAX_SIZE], int
result[MAX_SIZE][MAX_SIZE], int rowFirst, int
columnFirst, int rowSecond, int columnSecond)
```

```

{ // Initializing elements of result matrix to 0
for (int i = 0; i < rowFirst; ++i) {      for (int j = 0;
j < columnSecond; ++j) {                  result[i][j] = 0;
    }
}

```

```

// Multiplying firstMatrix and secondMatrix
and storing in result  for (int i = 0; i < rowFirst;
++i) {      for (int j = 0; j < columnSecond; ++j) {
for (int k = 0; k < columnFirst; ++k) {
    result[i][j] += firstMatrix[i][k] *
secondMatrix[k][j];
    }
}
}
}

```

```

void displayMatrix(int
matrix[MAX_SIZE][MAX_SIZE], int row, int column) {
for (int i = 0; i < row; ++i) {      for (int j = 0; j <
column; ++j) {          printf("%d\t", matrix[i][j]);
    }
    printf("\n");
}
}

```

```

int main() {    int firstMatrix[MAX_SIZE][MAX_SIZE],
secondMatrix[MAX_SIZE][MAX_SIZE],
result[MAX_SIZE][MAX_SIZE];    int rowFirst,
columnFirst, rowSecond, columnSecond;

    printf("Enter rows and columns for the first matrix:
");
    scanf("%d %d", &rowFirst, &columnFirst);

    printf("Enter elements of matrix 1:\n");
    for (int i = 0; i < rowFirst; ++i) {        for (int j = 0;
j < columnFirst; ++j) {            printf("Enter element
a%d%d: ", i + 1, j + 1);            scanf("%d",
&firstMatrix[i][j]);
        }
    }

    printf("Enter rows and columns for the second
matrix: ");    scanf("%d %d", &rowSecond,
&columnSecond);

    if (columnFirst != rowSecond) {        printf("Error!
Number of columns in the first matrix should be

```

equal to the number of rows in the second matrix.\n");

return 0;

}

printf("Enter elements of matrix 2:\n"); for
(int i = 0; i < rowSecond; ++i) { for (int j = 0; j <
columnSecond; ++j) { printf("Enter element
b%d%d: ", i + 1, j + 1); scanf("%d",
&secondMatrix[i][j]);

}

}

multiplyMatrices(firstMatrix, secondMatrix, result,
rowFirst, columnFirst, rowSecond, columnSecond);

// Displaying the multiplication result

printf("\nOutput Matrix:\n");

displayMatrix(result, rowFirst, columnSecond);

return 0;

}

Output:

```
C:\Users\Sarthak\Docun x + v
Enter rows and columns for the first matrix: 2 2
Enter elements of matrix 1:
Enter element a11: 1
Enter element a12: 2
Enter element a21: 3
Enter element a22: 4
Enter rows and columns for the second matrix: 2
2
Enter elements of matrix 2:
Enter element b11: 3
Enter element b12: 4
Enter element b21: 5
Enter element b22: 6

Output Matrix:
13      16
29      36

-----
Process exited after 23.51 seconds with return value 0
Press any key to continue . . . |
```

Week-9 Questions:

Q-1. Write a C program to

Search string. Ans-1.

```
#include <stdio.h> #include <string.h> int main()
```

```
{ char s1[] = "Beauty is in the eye of the
```

```

beholder"; char s2[] = "the"; int n = 0; int m = 0;
int times = 0; int len = strlen(s2); while(s1[n] !=
'\0') { if(s1[n] == s2[m]) { while(s1[n] == s2[m]
&& s1[n] != '\0') { n++; m++;
}
if(m == len && (s1[n] == ' ' || s1[n] == '\0')) { times++;
}
} else { while(s1[n] != ' ') { // Skip to
next word n++; if(s1[n] == '\0') break;
}
}
n++; m=0;
}
if(times > 0) { printf("'s' appears %d
time(s)\n", s2, times);
} else { printf("'s' does not appear in the
sentence.\n", s2);
}
return 0;
}

```

Output:


```
C:\Users\Sarthak\Docun x + v
'the' appears 2 time(s)

-----
Process exited after 10.05 seconds with return value 0
Press any key to continue . . . |
```

Q-2. Write a C program to Reverse words in string.

Ans-2.

```
#include <stdio.h>
#include <string.h>
#define MAX_SIZE 100
int main()
{
    char str[100], reverse[100]; int len,
    i, index, wordStart, wordEnd;
    printf("Enter any string: ");
    gets(str); len =
    strlen(str); index =
    0; wordStart = len -
```

```
1; wordEnd = len -
1; while(wordStart >
0)
{
if(str[wordStart] == ' ')
{
i = wordStart + 1; while(i
<= wordEnd)
{
reverse[index] =
str[i]; i++; index++;
}
reverse[index++] = ' ';
wordEnd = wordStart - 1;
}
wordStart--;
}
for(i=0; i<=wordEnd; i++)
{
reverse[index] = str[i];
index++;
```

```

}
reverse[index] = '\0'; printf("Original string
\n%s\n\n", str); printf("Reverse ordered
words \n%s", reverse); return 0;
}

```

Output:

```

C:\Users\Sarthak\Docun >
Enter any string: 2 4 6 8 9
Original string
2 4 6 8 9

Reverse ordered words
9 8 6 4 2
-----
Process exited after 14.36 seconds with return value 0
Press any key to continue . . .

```

Q-3. Write a C program to count vowels, consonants, etc.

Ans-3.

```

#include <stdio.h> int main() { char line[150];
int vowels, consonant, digit, space; vowels =
consonant = digit = space = 0; printf("Enter a
line of string: "); fgets(line, sizeof(line), stdin);
for (int i = 0; line[i] != '\0'; ++i) { line[i] =

```

```
tolower(line[i])); if (line[i] == 'a' || line[i] == 'e'
|| line[i] == 'i' || line[i] == 'o' || line[i] == 'u') {
++vowels;
} else if ((line[i] >= 'a' && line[i] <= 'z'))
{
++consonant;
}
else if (line[i] >= '0' && line[i] <= '9') {
++digit;
} else if (line[i] == ' ')
{
++space;
}
}

printf("Vowels: %d", vowels);
printf("\nConsonants: %d", consonant);
printf("\nDigits: %d", digit);
printf("\nWhite spaces: %d", space);
return 0;
}
```

Q-4. Create a program to separate characters in a given string?

Ans-4.

```
#include <stdio.h>
#include <stdlib.h>
void main(){
char str[100]; int l= 0; printf("\n\separate the
individual characters from a string :\n"); printf("-----
-----\n");
printf("Input the string : "); fgets(str, sizeof str,
stdin); printf("The characters of the string are : \n");
while(str[l]!='\0')
{
printf("%c ", str[l]);
l++;
}
printf("\n");
}
```

Q-5. Write a program to take two strings from user and concatenate

them also add a space between them using strcat() function.

Ans-5.

```
#include <stdio.h>
#include <string.h>
int main()
{
char a[100], b[100]; printf("Enter the first string\n");
gets(a); printf("Enter the second string\n"); gets(b);
strcat(a,b); printf("String obtained on concatenation
is %s\n",a); return 0;
}
```

Sample Input:

Jai Gla

Jai Gla

Sample output:

Identical

Q-6. Write a C program to take a string from user and make it toggle its case

i.e. lower case to upper case and upper case to lower case.

Ans-6.

```
#include <stdio.h> void
toggleChars(char str[])
{ for (int i = 0; str[i] != '\0'; i++) {
if (str[i] >= 'A' && str[i] <= 'Z')
str[i] = str[i] + 'a' - 'A'; else if
(str[i] >= 'a' && str[i] <= 'z') str[i]
= str[i] + 'A' - 'a';
}
}

int main()
{
char str[] = "HElLo wOrlD";
toggleChars(str); printf("String after toggle
\n"); printf("%s\n", str); return 0;
}
```

Q-7. Write a C program to take two strings as input from user and check

they are identical or not without using string functions.

Ans-7.

```
#include <stdio.h>
#include <string.h>
int main()
{
    char Str1[100], Str2[100]; int result, i;
    printf("\n Please Enter the First String : ");
    gets(Str1); printf("\n Please Enter the
    Second String : "); gets(Str2);

    for(i = 0; Str1[i] == Str2[i] && Str1[i] != '\0'; i++);
    if(Str1[i] < Str2[i])
    {
        printf("\n str1 is Less than str2");
    } else if(Str1[i] >
    Str2[i])
    {
        printf("\n str2 is Less than str1");
    }
}
```



```
else
{
printf("\n str1 is Equal to str2");
}
return 0;
}
```

Q-8. Write a C program to take a list of a student's names from user by asking number of students and sort them alphabetical order.

Ans-8.

```
#include <stdio.h>
#include <stdlib.h> #include <string.h> int
main() { int numStudents; printf("Enter the
number of students: "); scanf("%d",
&numStudents); if (numStudents <= 0) {
printf("Invalid number of students.
Exiting...\n"); return 1;
}
```

```

char **studentNames = (char **)malloc(numStudents
* sizeof(char *)); for (int i = 0; i < numStudents; i++) {
printf("Enter the name of student %d: ", i + 1);
studentNames[i] = (char *)malloc(100 *
sizeof(char)); scanf("%s", studentNames[i]);

}
for (int i = 0; i < numStudents - 1; i++) { for (int j = i
+ 1; j < numStudents; j++) { if
(strcmp(studentNames[i], studentNames[j]) > 0) {
char *temp = studentNames[i]; studentNames[i] =
studentNames[j]; studentNames[j] = temp;
}
}
}

printf("\nSorted names in alphabetical
order:\n"); for (int i = 0; i < numStudents; i++) {
printf("%d. %s\n", i + 1, studentNames[i]);
}

for (int i = 0; i < numStudents; i++) {
free(studentNames[i]);
}

```

```
free(studentNames);  
return 0; }
```

Week-10 Questions:

Q-1. Write a C program to find length of string using pointers.

Ans-1.

```
#include <stdio.h> int  
main() { char str[100], *  
ptr; int count;  
printf("Enter any string:  
"); gets(str);  
  
ptr = str; count  
= 0;  
  
while ( *ptr != '\0') {  
count++; ptr++;  
}  
printf("The length of the string is: %d", count);  
return 0;
```

```
}
```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter any string: G L A U N I V E R S I T Y
The length of the string is: 25
-----
Process exited after 37.42 seconds with return value 0
Press any key to continue . . . |
```

Q-2. Write a C program to copy one string to another using pointer.

Ans-2.

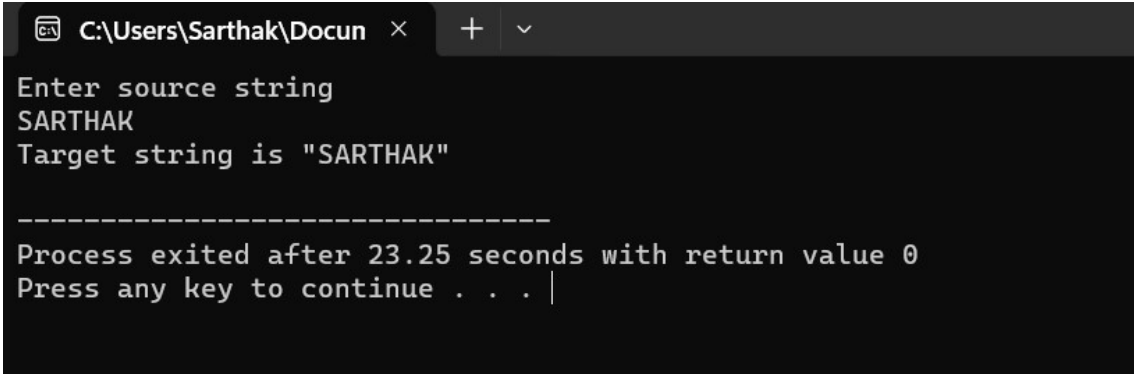
```
#include<stdio.h> void
copy_string(char*, char*);
main()
{
char source[100], target[100];
printf("Enter source string\n");
gets(source); copy_string(target, source);
printf("Target string is \"%s\"\n", target);
return 0;
}
void copy_string(char *target, char *source)
```

```

{
while(*source)
{
*target = *source;
source++;
target++;
}
*target = '\0';
}

```

Output:



```

C:\Users\Sarthak\Docun >
Enter source string
SARTHAK
Target string is "SARTHAK"

-----
Process exited after 23.25 seconds with return value 0
Press any key to continue . . . |

```

Q-3 Write a C program to concatenate two strings using pointers.

Ans-3.

```

#include <stdio.h> void
concatenate(char *str1, char *str2) {
while (*str1) { str1++;

```

```
}  
while (*str2) {  
    *str1 = *str2;  
    str1++; str2++;  
}  
*str1 = '\\0';  
}  
  
int main() { char string1[100],  
string2[50]; printf("Enter the  
first string:\\n"); gets(string1);  
  
printf("Enter the second string:\\n");  
gets(string2); concatenate(string1, string2);  
printf("Concatenated string: %s\\n",  
string1); return 0;  
}
```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter the first string:
C O M P U T E R
Enter the second string:
E N G I N E E R I N G
Concatenated string: C O M P U T E R E N G I N E E R I N G

-----
Process exited after 40.44 seconds with return value 0
Press any key to continue . . . |
```

Q-4. Write a C program to compare two strings using pointers.

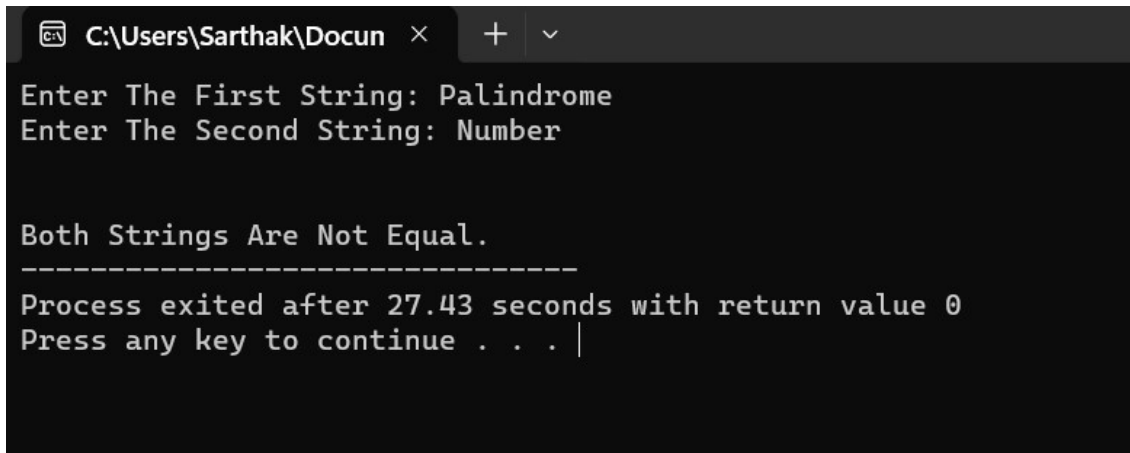
Ans-4.

```
#include <iostream> using
namespace std; int main()

{
char
string1[50],string2[50],*str1,*str2; int
i,equal = 0; printf("Enter The First
String: "); scanf("%s",string1);
printf("Enter The Second String: ");
scanf("%s",string2); str1 = string1; str2
= string2; while(*str1 == *str2)
{
```

```
if ( *str1 == '\0' || *str2 == '\0' )  
    break; str1++; str2++;  
}  
if( *str1 == '\0' && *str2 == '\0' )  
    printf("\n\nBoth Strings Are Equal.");  
else  
  
    printf("\n\nBoth Strings Are Not Equal.");  
}
```

Output:



```
C:\Users\Sarthak\Docun × + v  
Enter The First String: Palindrome  
Enter The Second String: Number  
  
Both Strings Are Not Equal.  
-----  
Process exited after 27.43 seconds with return value 0  
Press any key to continue . . . |
```

Q-5 WAP to find largest among three numbers using pointer.

Q-6 WAP to find largest among three numbers using pointer.

Ans-5 & 6.


```
#include<stdio.h> int
main()
{
int a,b,c,*pa, *pb, *pc;
printf("Enter three numbers:\n");
scanf("%d%d%d", &a,&b,&c); /*
Referencing */ pa= &a; pb= &b; pc=
&c; if(*pa > *pb && *pa > *pc)
{
printf("Largest is: %d", *pa);
}
else if(*pb > *pc && *pb > *pc)
{
printf("Largest is : %d", *pb);
}
else
{
printf("Largest = %d", *pc);
}
return 0;
}
```

Output:

```
C:\Users\Sarthak\Docun x + v
Enter three numbers:
4 6 8
Largest = 8
-----
Process exited after 14.9 seconds with return value 0
Press any key to continue . . . |
```

Q-7. WAP to find factorial of a number using pointer.

Ans-7.

```
#include<stdio.h> void findFactorial(int,int
```

```
*); int main(){ int i,factorial,n;
```

```
printf("Enter a number: ");
```

```
scanf("%d",&n);
```

```
findFactorial(n,&factorial);
```

```
printf("Factorial of %d is: %d",n,*factorial);
```

```
return 0;
```

```
}
```

```
void findFactorial(int n,int *factorial){
```

```
int i;
```

```
*factorial =1; for(i=1;i<=n;i++)
```

```
*factorial=*factorial*i;  
}
```

Q-8. Write a program to print largest even number present in an array using pointer to an array.

Ans-8.

```
#include <stdio.h> int findLargestEven(int  
*arr, int size) { int largestEven = -1; for  
(int i = 0; i < size; i++) { if (arr[i] % 2 == 0  
&& arr[i] > largestEven) { largestEven =  
arr[i];  
}  
}  
return largestEven;  
}  
  
int main() { int size; printf("Enter the  
size of the array: "); scanf("%d",  
&size); int arr[size]; printf("Enter %d  
elements:\n", size); for (int i = 0; i <
```

```

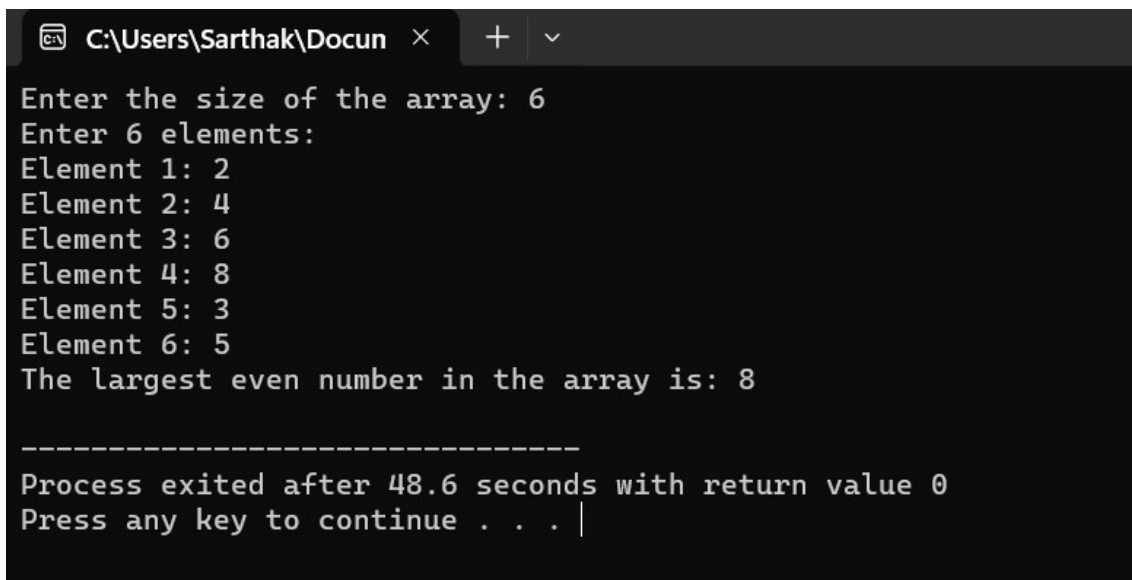
size; i++) { printf("Element %d: ", i +
1); scanf("%d", &arr[i]);
}

int *ptr = arr; int largestEven = findLargestEven(ptr,
size); if (largestEven != -1) { printf("The largest even
number in the array is: %d\n", largestEven);
} else { printf("No even numbers found in the
array.\n");
}

return 0;
}

```

Output:



```

C:\Users\Sarthak\Docum x + v
Enter the size of the array: 6
Enter 6 elements:
Element 1: 2
Element 2: 4
Element 3: 6
Element 4: 8
Element 5: 3
Element 6: 5
The largest even number in the array is: 8

-----
Process exited after 48.6 seconds with return value 0
Press any key to continue . . .

```

Q-9. WAP to find sum of elements of an array using array of pointer.

Ans-9.

```
#include <stdio.h>

#include <malloc.h> void

main()

{

int i, n, sum = 0; int *a; printf("Enter
the size of array A \n"); scanf("%d",
&n);

a = (int *) malloc(n * sizeof(int));
printf("Enter Elements of the List \n");
for (i = 0; i < n; i++)
{
scanf("%d", a + i);
} for (i = 0; i < n;
i++)
{
sum = sum + *(a + i); printf("Sum of all elements
in array = %d\n", sum); return 0;
}
```

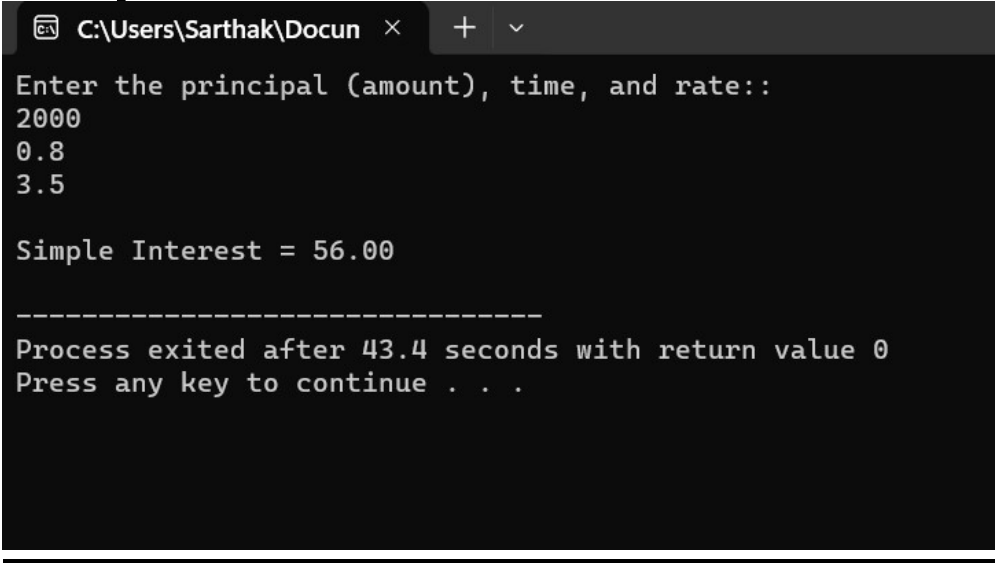
Q-10. WAP to compute simple interest using pointers.

Ans-10.

```
#include<stdio.h> int main() { float p, t, r, SI;
float *x, *y, *z; // These are the pointer
variables printf("Enter the principal (amount),
time, and rate::\n"); scanf("%f%f%f", &p, &t,
&r); x = &p; y = &t; z = &r;

SI = (*x * *y * *z) / 100; printf("\nSimple
Interest = %.2f\n", SI); return 0;
}
```

Output:



```
C:\Users\Sarthak\Docum x + v
Enter the principal (amount), time, and rate::
2000
0.8
3.5

Simple Interest = 56.00

-----
Process exited after 43.4 seconds with return value 0
Press any key to continue . . .
```

Q-11. Write a program to print largest even number present in an array using pointer to an array.

Ans-11.

```
#include <stdio.h> int findLargestEven(int *arr, int  
size) { int largestEven = -1; // Assuming all elements  
are nonnegative
```

```
for (int i = 0; i < size; i++) { if (arr[i] % 2 ==  
0 && arr[i] > largestEven) { largestEven =  
arr[i];
```

```
}
```

```
}
```

```
return largestEven;
```

```
}
```

```
int main() { int
```

```
size;
```

```
printf("Enter the size of the array: ");
```

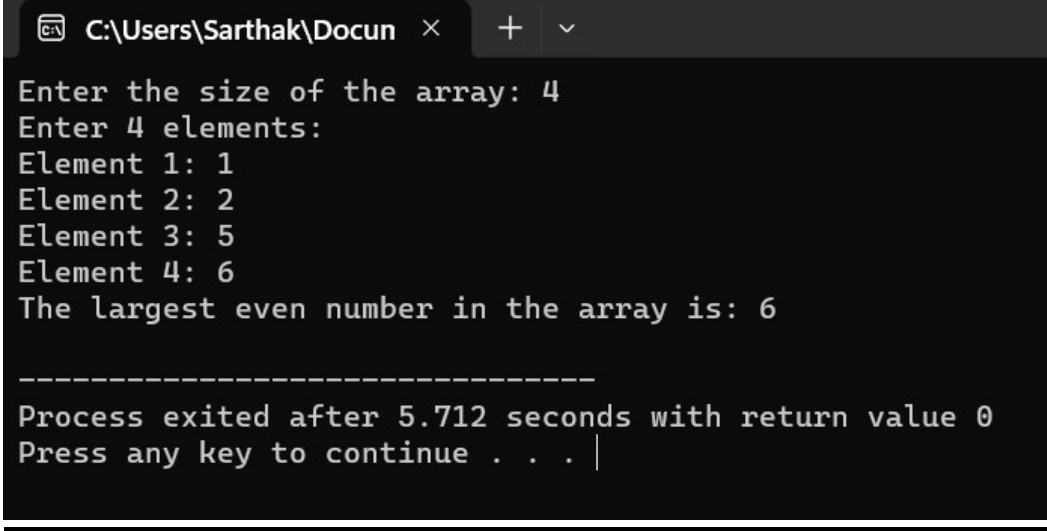
```
scanf("%d", &size); int arr[size];
```

```
printf("Enter %d elements:\n",
```

```
size); for (int i = 0; i < size; i++) {
```

```
printf("Element %d: ", i + 1);  
scanf("%d", &arr[i]);  
}  
  
int *ptr = arr; int largestEven = findLargestEven(ptr,  
size); if (largestEven != -1) { printf("The largest even  
number in the array is: %d\n", largestEven);  
} else { printf("No even numbers found in the  
array.\n");  
}  
  
return 0;  
}
```

Output:



```
C:\Users\Sarthak\Docun × + v  
Enter the size of the array: 4  
Enter 4 elements:  
Element 1: 1  
Element 2: 2  
Element 3: 5  
Element 4: 6  
The largest even number in the array is: 6  
  
-----  
Process exited after 5.712 seconds with return value 0  
Press any key to continue . . . |
```


Week-11 Questions:

Q-1. Write a C function to return the maximum of three integers.

Ans-1.

```
#include<stdio.h> double max3(double  
x,double y,double z); void main () {  
double i; double a,b,c; clrscr();  
printf("Enter the value of x,y,z:\n");  
scanf("%lf%lf%lf",&a,&b,&c); i=  
max3(a,b,c) ; printf("%lf",i); getch();  
}
```

```
double max3(double x,double y,double z) {
double max;

if (x > y) max
= x; else max
= y; if (z >
max) max =
z; return
max;
}
```

Q-2. Write a C function to check if a given number is prime or not.

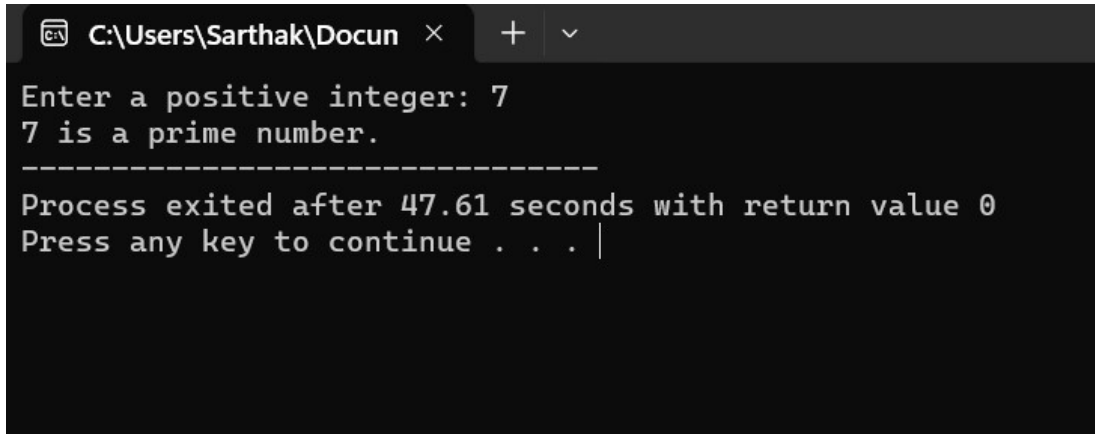
Ans-2.

```
#include <stdio.h> int main() { int
n, i, flag = 0; printf("Enter a
positive integer: "); scanf("%d",
&n); if (n == 0 || n == 1) flag = 1;
for (i = 2; i <= n / 2; ++i) { if (n % i
== 0) {

flag = 1; break;
}
```

```
} if (flag == 0) printf("%d is a prime  
number.", n); else printf("%d is not a  
prime number.", n); return 0;  
}
```

Output:



```
C:\Users\Sarthak\Docun > Enter a positive integer: 7  
7 is a prime number.  
-----  
Process exited after 47.61 seconds with return value 0  
Press any key to continue . . . |
```

Q-3 Write a C function to compute the factorial of a non-negative integer.

Ans-3.

```
#include <stdio.h> unsigned long  
long factorial(int n) {    if (n == 0 ||  
n == 1) {        return 1;    } else {  
return n * factorial(n - 1);  
    }  
}
```

```

int main() {    int num;    printf("Enter a non-
negative integer: ");    scanf("%d", &num);    if
(num < 0) {        printf("Factorial is not defined
for negative numbers.\n");

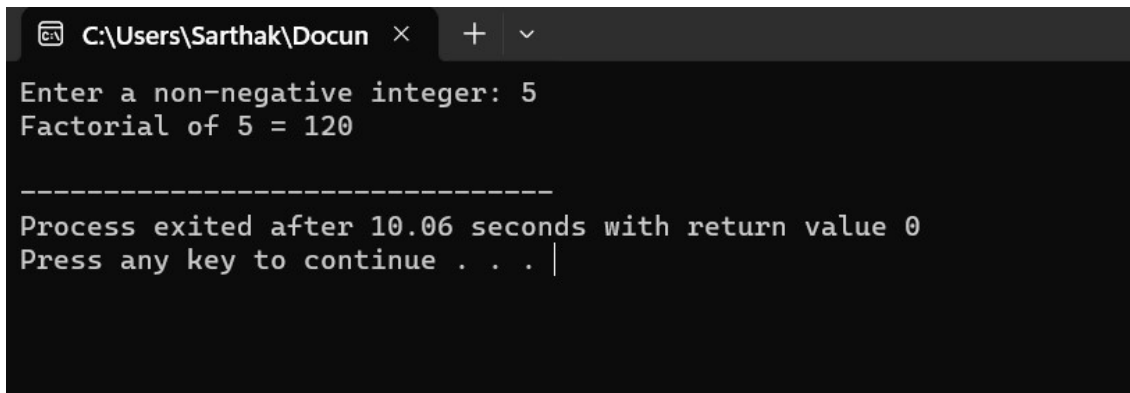
        } else {        unsigned long long result =
factorial(num);        printf("Factorial of %d =
%llu\n", num, result);

        }

    return 0;
}

```

Output:



```

C:\Users\Sarthak\Docun >
Enter a non-negative integer: 5
Factorial of 5 = 120

-----
Process exited after 10.06 seconds with return value 0
Press any key to continue . . . |

```

Q-4 Write a C function to swap the values of two integers in actual arguments.

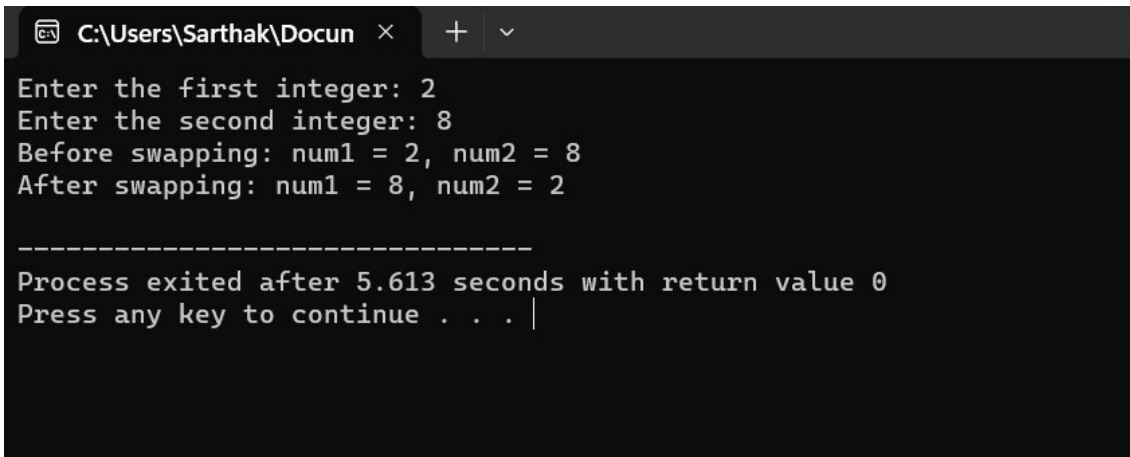
Ans-4.

```
#include <stdio.h> void
swapIntegers(int *a, int *b) {
    int temp = *a;    *a = *b;
        *b = temp;
}

int main() {    int
num1, num2;
printf("Enter the
first integer: ");
scanf("%d",
&num1);
printf("Enter the
second integer: ");
scanf("%d",
&num2);
printf("Before
swapping: num1 =
%d, num2 =
%d\n", num1,
num2);
swapIntegers(&nu
```

```
m1, &num2);  
printf("After  
swapping: num1 =  
%d, num2 =  
%d\n", num1,  
num2);  
    return 0;  
}
```

Output:



```
C:\Users\Sarthak\Docum × + v  
Enter the first integer: 2  
Enter the second integer: 8  
Before swapping: num1 = 2, num2 = 8  
After swapping: num1 = 8, num2 = 2  
  
-----  
Process exited after 5.613 seconds with return value 0  
Press any key to continue . . . |
```

Q-5. Write a C function to compute the sum and average of an array of integers.

Ans-5.

```
#include <stdio.h> int main(){ int  
arr[100], size, sum; float avg;
```

```
printf("Enter the size of the array:
"); scanf("%d", &size); printf("Enter
the array elements: "); for(int i = 0; i
< size; i++){ scanf("%d", &arr[i]);
}
sum = 0;
for(int i = 0; i < size; i++){
sum = sum + arr[i];
}
avg = sum / size;
printf("Sum of array elements is: %d", sum);
printf("\nAvg. of arrays elements is: %.2f", avg);
return 0;
}
```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter the size of the array: 5
Enter the array elements: 2
4
6
8
0
Sum of array elements is: 20
Avg. of arrays elements is: 4.00
-----
Process exited after 15.42 seconds with return value 0
Press any key to continue . . . |
```

Q-6. Write a C function to find the GCD (Greatest Common Divisor) of two non negative integers using Euclid's algorithm.

Ans-6.

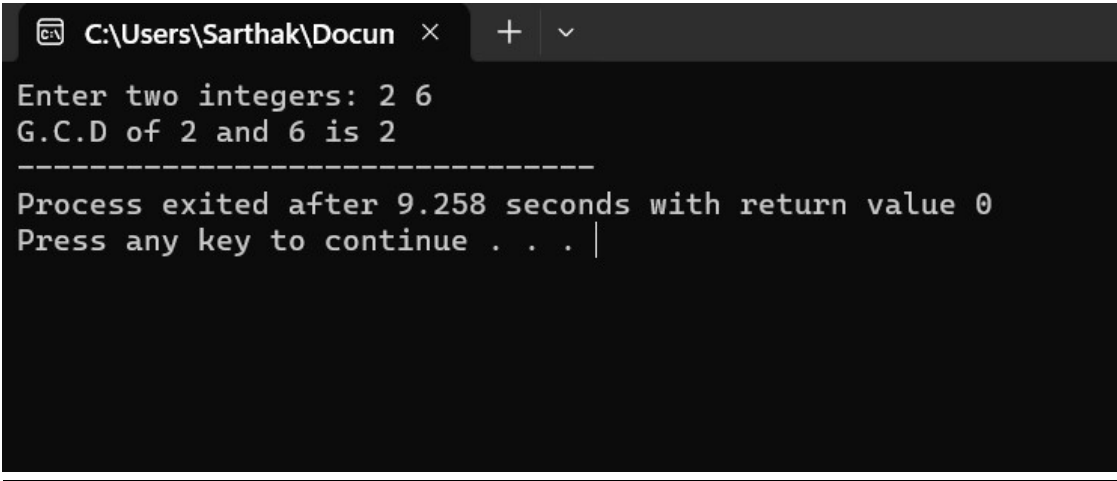
```
#include <stdio.h> int
main()

{
int n1, n2, i, gcd; printf("Enter
two integers: "); scanf("%d
%d", &n1, &n2); for(i=1; i <=
n1 && i <= n2; ++i)
{
```



```
if(n1%i==0 && n2%i==0)
gcd = i;
}
printf("G.C.D of %d and %d is %d", n1, n2, gcd);
return 0;
}
```

Output:



```
C:\Users\Sarthak\Docun >
Enter two integers: 2 6
G.C.D of 2 and 6 is 2
-----
Process exited after 9.258 seconds with return value 0
Press any key to continue . . . |
```

Q-7. Write a C function to check if a given string is a valid palindrome, considering only alphanumeric characters and ignoring cases.

Ans-7.

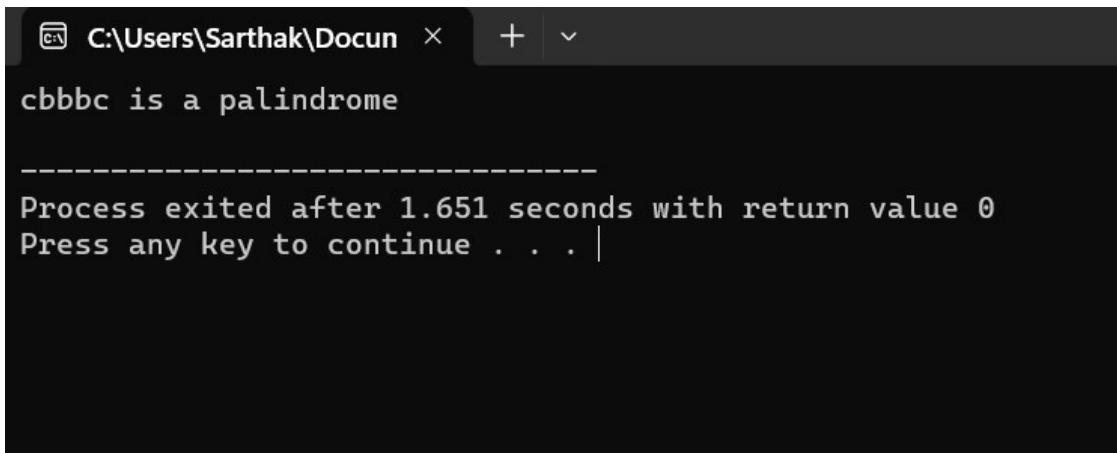
```
#include <stdio.h>
#include <string.h>
int main()
```

```

{
char str[] = { "cbbbc" }; int l = 0; int h =
strlen(str) - 1; while (h > l) { if (str[l++]
!= str[h--]) { printf("%s is not a
palindrome\n", str); return 0;
}
}
printf("%s is a palindrome\n", str);
return 0;
}

```

Output:



```

C:\Users\Sarthak\Docun >
cbbbc is a palindrome
-----
Process exited after 1.651 seconds with return value 0
Press any key to continue . . . |

```

Q-8. Write a C function to calculate the sum and difference of two complex numbers.

Ans-8.

```
#include <stdio.h> void addComplex(float real1,
float imaginary1, float real2, float imaginary2, float
*resultReal, float *resultImaginary) {
    *resultReal = real1 + real2;
    *resultImaginary = imaginary1 + imaginary2;
}
```

```
int main() {    float real1, imaginary1, real2,
imaginary2;    printf("Enter the real and imaginary
parts of the first complex number: ");    scanf("%f %f",
&real1, &imaginary1);    printf("Enter the real and
imaginary parts of the second complex number: ");
scanf("%f %f", &real2, &imaginary2);    float sumReal,
sumImaginary;    addComplex(real1, imaginary1,
real2, imaginary2,
&sumReal, &sumImaginary);    printf("Sum: %.2f
+ %.2fi\n", sumReal, sumImaginary);
printf("Difference: %.2f + %.2fi\n", real1 - real2,
imaginary1 - imaginary2);

    return 0;
}
```

Output:

```
C:\Users\Sarthak\Docun × + v
Enter the real and imaginary parts of the first complex number: 2
8
Enter the real and imaginary parts of the second complex number: 4
2
Sum: 6.00 + 10.00i
Difference: -2.00 + 6.00i

-----
Process exited after 18.66 seconds with return value 0
Press any key to continue . . . |
```

```
C:\Users\Sarthak\Docun × + v
Enter the real and imaginary parts of the first complex number: 2
8
Enter the real and imaginary parts of the second complex number: 4
2
Sum: 6.00 + 10.00i
Difference: -2.00 + 6.00i

-----
Process exited after 18.66 seconds with return value 0
Press any key to continue . . . |
```