

Name: Makwana Rudra Ketambhai

Roll: 25BCL131

Sub: C Programming Assignment

Unit: 2 Arrays And Strings

Section A: Programming Using Arrays

1 Accept 5 values and print them later on.

⇒ #include <stdio.h>

```
int main() {  
    int arr[5], i;
```

//Accept 5 values

```
printf("Enter 5 values: \n");
```

```
for (i = 0; i < 5; i++) {
```

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

}

//Print the values

```
printf("You entered: \n");
```

```
for (i = 0; i < 5; i++) {
```

```
    printf("%d", arr[i]);
```

}

```
return 0;
```

}

Output: Enter 5 values:-

1

2

3

4

5

The accepted value are :- 1 2 3 4 5

2. Accept 10 values and print 4th, 7th, and 9th value

=> ~~#include <stdio.h>~~

```
int main() {
    int arr[10], i;
    printf ("Enter 10 values: ");
    for (i = 0; i < 10; i++) {
        scanf ("%d", &arr[i]);
    }
    printf ("4th value = %d\n", arr[3]);
    printf ("7th value = %d\n", arr[6]);
    printf ("9th value = %d\n", arr[8]);
}
return 0;
}
```

Output: Enter 10 integer values :-

Enter value 1: 1

Enter value 2: 2

Enter value 3: 3

Enter value 4: 4

Enter value 5: 5

Enter value 6: 6

Enter value 7: 7

Enter value 8: 8

Enter value 9: 9

Enter value 10: 10

3. Accept 5 values and sorting according /descending

=> #include < stdio.h >

```
int main() {  
    int arr[5], i, j, temp;  
    printf ("Enter 5 values: ");  
    for(i=0; i<5; i++)  
        scanf ("%d", &arr[i]);
```

// Ascending order

```
for(i=0; i<5; i++) {  
    for(j=i+1; j<5; j++) {  
        if(arr[i] > arr[j]) {  
            temp = arr[i];  
            arr[i] = arr[j];  
            arr[j] = temp;  
        }  
    }  
}
```

```
printf ("Ascending order: ");  
for (i = 0; i < 5; i++)  
    printf ("%d", arr[i]);
```

// Descending order

```
printf ("In Descending order: ");  
for (i = 4; i >= 0; i--)  
    printf ("%d", arr[i]);  
return 0;
```

Output : Enter 5 integer values:

7 8 10 18 45

Enter 1 for ascending order, 0 for descending order: 0

Sorted array: 45 18 10 8 7

4. Print minimum number of notes required for a given amount

=> #include <stdio.h>

```
int main() {
    int amount, notes[9] = {500, 200, 100,
                           50, 20, 10, 5, 2, 1};
    int i, count;
    printf("Enter the amount:");
    scanf("%d", &amount);
    for (i = 0; i < 9; i++) {
        count = amount / notes[i];
        amount = amount % notes[i];
        if (count != 0)
            printf("%d x %d\n", notes[i],
                   count);
    }
    return 0;
}
```

Output: for amount 1024:
2 x 500
1 x 20
2 x 25 of 8E-14 : Xxxx

Q5. Add two 2D arrays of same size and store result in third

=> #include <stdio.h>

```
int main() {
    int a[2][2], b[2][2], sum[2][2];
    int i, j;
    printf("Enter elements of first 2x2
matrix:\n");
    for(i=0; i<2; i++)
        for(j=0; j<2; j++)
            scanf("%d", &a[i][j]);
    printf("Enter elements of second 2x2
matrix:\n");
    for(i=0; i<2; i++)
        for(j=0; j<2; j++)
            scanf("%d", &b[i][j]);
    for(i=0; i<2; i++)
        for(j=0; j<2; j++)
            sum[i][j] = a[i][j] + b[i][j];
}
```

```

    printf ("sum of two matrices: \n");
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 2; j++)
            printf ("%d", sum[i][j]);
        printf ("\n");
    }
    return 0;
}

```

Output: Sum of the two 2D arrays :

10 10 10

10 10 10

10 10 10

6. Multiply two 2D arrays and store result in third

⇒ #include <stdio.h>

```

int main() {
    int a[2][2], b[2][2], m[2][2];
    int i, j, k;
    printf ("Enter first 2x2 matrix: \n");
    for (i = 0; i < 2; i++)
        for (j = 0; j < 2; j++)
            scanf ("%d", &a[i][j]);
    printf ("Enter second 2x2 matrix: \n");
    for (i = 0; i < 2; i++)
        for (j = 0; j < 2; j++)
            scanf ("%d", &b[i][j]);
    for (i = 0; i < 2; i++)
        for (j = 0; j < 2; j++)
            m[i][j] = a[i][j] * b[i][j];
    printf ("Product matrix: \n");
    for (i = 0; i < 2; i++)
        for (j = 0; j < 2; j++)
            printf ("%d ", m[i][j]);
    return 0;
}

```

```
printf("Enter second 2x2 matrix:\n");
for(i=0; i<2; i++)
```

```
    for(j=0; j<2; j++)
```

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

```
for(i=0; i<2; i++) {
```

```
    for(j=0; j<2; j++) {
```

```
        m4[i][j] = 0;
```

```
        for(k=0; k<2; k++)
```

```
            m4[i][j] += a[i][k] * b[k][j];
```

```
}
```

```
printf("Multiplication result:\n");
```

```
for(i=0; i<2; i++) {
```

```
    for(j=0; j<2; j++)
```

```
        printf("%d ", m4[i][j]);
```

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

```
}
```

```
return 0;
```

Output :- Enter elements for first matrix (3x2):

Enter elements matrix 1 [0][0]: 0

Enter elements matrix 1 [0][1]: 1

Enter elements matrix 1 [1][0]: 2

Enter elements matrix 1 [1][1]: 3

Enter elements matrix 1 [2][0]: 4

Enter elements matrix 1 [2][1]: 5

Enter elements for second matrix (2x3):
Enter elements matrix 2 [0][0]: 6
Enter elements matrix 2 [0][1]: 7
Enter elements matrix 2 [0][2]: 8
Enter elements matrix 2 [1][0]: 9
Enter elements matrix 2 [1][1]: 10
Enter elements matrix 2 [1][2]: 11

Resultant matrix after multiplication (3x3):

9 10 11

39 44 49
69 78 87

7 Obtain transpose of a 4x4 matrix

```
#include <stdio.h>
int main()
{
    int a[4][4], t[4][4];
    int i, j;
    printf("Enter 4x4 matrix:\n");
    for(i=0; i<4; i++)
        for(j=0; j<4; j++)
            scanf("%d", &a[i][j]);
    for(i=0; i<4; i++)
        for(j=0; j<4; j++)
            t[j][i] = a[i][j];
```

```

    printf("Transpose of matrix:\n");
    for(i=0; i < 4; i++) {
        for(j=0; j < 4; j++) {
            printf("%d", t[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

Output: Enter elements of 4x4 matrix:

Enter elements matrix [0][0]: 0

Enter elements matrix [0][1]: 1

Enter elements matrix [0][2]: 2

Enter elements matrix [0][3]: 3

Enter elements matrix [1][0]: 4

Enter elements matrix [1][1]: 5

Enter elements matrix [1][2]: 6

Enter elements matrix [1][3]: 7

Enter elements matrix [2][0]: 8

Enter elements matrix [2][1]: 9

Enter elements matrix [2][2]: 10

Enter elements matrix [2][3]: 11

Enter elements matrix [3][0]: 12

Enter elements matrix [3][1]: 13

Enter elements matrix [3][2]: 14

Enter elements matrix [3][3]: 15

Original Matrix:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Transposed Matrix:

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

8 (Copy one array of 5 elements to another array of 10 (skipping some position))

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

```
int main() {
    int a[5], b[10];
    int i, j = 0;
    printf("Enter 5 elements: ");
    for (i = 0; i < 5; i++) {
        scanf("%d", &a[i]);
    }
    for (i = 0; i < 10; i++) {
        if (j > 2 == 0)
            b[i] = a[j++];
        else
            b[i] = 0;
    }
}
```

```
printf("New array:\n");
for (i = 0; i < 10; i++)
    printf("%d", b[i]);
return 0;
```

Output:

Source Array: 10 20 30 40 50

Destination Array: 10 0 20 0 30 0 40 0 50 0

9 Reverse an array of maximum 5 elements

⇒ #include < stdio.h >

```
int main() {
    int arr[5], i;
    printf("Enter 5 elements: ");
    for (i = 0; i < 5; i++)
        scanf("%d", &arr[i]);
    printf("Reversed array: \n");
    for (i = 4; i >= 0; i--)
        printf("%d", arr[i]);
    return 0;
}
```

Output:

Enter the number of elements (maximum 5):

Enter 3 elements:

Element 1: 7

Element 2: 18

Element 3: 45

Original array : 7 18 45
reversed array : 45 18 7

Q Find frequency of each number in an array (20 elements)

⇒ #include <stdio.h>

```
int main() {
    int arr[20], freq[20];
    int i, j, count;
    printf("Enter 20 elements:");
    for (i = 0; i < 20; i++) {
        scanf("%d", &arr[i]);
        freq[i] = -1;
    }
    for (i = 0; i < 20; i++) {
        count = 1;
        for (j = i + 1; j < 20; j++) {
            if (arr[i] == arr[j]) {
                count++;
                freq[i] = 0;
            }
        }
        if (freq[i] != 0)
            freq[i] = count;
    }
}
```

```

printf("Frequency of each element:\n");
for(i=0; i < 10; i++) {
    if(freq[i] != 0)
        printf("%d occurs %d times\n",
               arr[i], freq[i]);
}
return 0;
}

```

Output:

Element	Frequency
1	2
2	3
8	3
3	1
5	1

\Rightarrow Shift all numbers by n positions within an array of 10 elements (left or right), pad with 0.

\Rightarrow #include <stdio.h>

```

int main() {
    int arr[10], result[10];
    int i, n, dir;
    // Input array elements

```

```
printf("Enter 20 numbers:\n");
for(i=0; i<20; i++) {
    scanf("%d", &arr[i]);
}
```

```
// Input shift value
printf("Enter number of positions to
shift: ");
scanf("%d", &n);
```

```
// Input direction (1 = left, 2 = right)
printf("Enter direction (1 for left, 2
for right): ");
scanf("%d", &dir);
```

```
// Initialize result array with zeros
for(i=0; i<20; i++) {
    result[i] = 0;
}
```

```
// Perform shift
```

```
if(dir == 1) { // Left shift
    for(i=0; i<20-n; i++) {
        result[i] = arr[i+n];
    }
}
```

```
} else if (dir == 2) { // Right shift
    for(i=n; i<20; i++) {
        result[i] = arr[i-n];
    }
}
```

```
} else {
```

```

        printf ("Invalid direction!\n");
        return 0;
    }

    // Print shifted array
    printf ("Shifted array:\n");
    for (i = 0; i < 10; i++) {
        printf ("%d", result[i]);
    }
}

```

return 0;

Output

- * Entered 10 numbers:

=> 1 2 3 4 5 6 7 8 9 10

- * Array after 3 position right shift:

=> 0 0 0 1 2 3 4 5 6 7

- * Array after 2 position left shift:

3 4 5 6 7 8 9 10 0 0

J2 Insert a new number at the beginning of the array.

=> #include < stdio.h >

int main()

: int arr[100] = {1, 2, 3, 4, 5};

int size = 5;

int new number = 0;

```

printf ("Original array:");
for (int i = 0; i < size; i++) {
    printf ("%d", arr[i]);
}
printf ("\n");
for (int i = size - 1; i >= 0; i--) {
    arr[i + 1] = arr[i];
}
arr[0] = new number;
size++;
printf ("Array after insertion ");
for (int i = 0; i < size; i++) {
    printf ("%d", arr[i]);
}
printf ("\n");
return 0;
}

```

Output: Original array: 1 2 3 4 5
 Array after insertion: 0 1 2 3 4 5

=> Insert a new number at a particular position in an array

=> #include <stdio.h>
int main() {

```

int array [100], position, n, value;
printf ("Enter number of element in array\n");
scanf ("%d", &n);

```

```
printf("Enter no. of elements\n", n);
for (c=0; c < n; c++)
    scanf("%d", &array[c]);
printf("Enter the location where you wish to insert an element\n");
scanf("%d", &position);
printf("Enter the value to insert\n");
scanf("%d", &value);
for (c=n-1; c >= position-1; c--)
    array[c+1] = array[c];
array[position-1] = value;
printf("Resultant array is\n");
for (c=0; c <= n; c++)
    printf("%d\n", array[c]);
return 0;
```

Output:

Enter number of elements in array : 7

Enter 7 elements : 7

Enter the location where you wish to insert an element : 3

Enter the value to insert : 8

Resultant array is

8

7

34 Insert a new number at the last position
of an array.

```
=> #include <stdio.h>
int main() {
    int arr[100] = {1, 2, 3, 4, 5, 6};
    int size = 6;
    int new-number = 7;
    if (size < 100) {
        arr[size] = new-number;
        size++;
        printf ("Array after insertion:");
        for (int i = 0; i < size; i++) {
            printf ("%d", arr[i]);
        }
        printf ("\n");
    } else {
        printf ("Array is full cannot insert
new number.\n");
    }
    return 0;
}
```

Output: Array after insertion: 1 2 3 4 5 6

35 Delete a value from the first position of an array.

```
=> #include <stdio.h>
int main() {
    int arr[100];
```

```

int size, i;
printf ("Enter the size of the array:");
scanf ("%d", &size);

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

printf ("Array after deleting the first element:");
for (i = 0; i < size; i++) {
    printf ("%d", arr[i]);
}
printf ("\n");
return 0;
}

```

Output:-

Enter the size of the array: 2
 Enter the elements of the arrays: 18 7
 Array after deleting the first element: -7

- J6 Delete a value from a particular position in an array.

$\Rightarrow \#include <stdio.h>$

```
int main() {
    int arr[700], n, pos, i;
    printf("Enter number of elements:");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter position to delete (0 to %d):\n", n);
    scanf("%d", &pos);

    if (pos < 0 || pos > n) {
        printf("Invalid position.\n");
        return 0;
    }

    // Shift elements to left from the position
    for (i = pos - 1; i < n - 1; i++)
        arr[i] = arr[i + 1];
    n--; // Reduce array size

    printf("Array after deletion:\n");
    for (i = 0; i < n; i++)
        printf("%d", arr[i]);
}

return 0;
```

Output:

Enter the size of the array : 2
Enter 2 elements in the array : 28 7
Enter the position of the element to delete : 1
Array after deletion : 7

I7: Delete a value from the last position of an array.

=> #include <stdio.h>

int main()

int arr[100], n;

printf("Enter number of elements:");

scanf("%d", &n);

if (n <= 0) {

printf("Array is empty.\n");

}

printf("Enter %d elements: ", n);

for (int i = 0; i < n; i++)

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

n -= 1; //Delete last element

printf("Updated array: ");

for (int i = 0; i < n; i++)

printf("%d", arr[i]);

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

```
    return 0;
```

Output: Enter number of elements : 5

Enter elements : 1 2 3 4 5

Updated array : 1 2 3 4

J8 Delete a value from the array (search and remove):

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

```
int main() {
    int arr[100], n, i, j, value, found = 0;
    printf("Enter the number of element in the
array:");
    scanf("%d", &n);
    printf("Enter %d elements: \\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    printf("Enter the value to delete:");
    scanf("%d", &value);
    // search and delete
    for (i = 0; i < n; i++) {
        if (arr[i] == value) {
            found = 1;
            for (j = i; j < n - 1; j++) {
```

array[j] = array[j+1]; // shift
elements left

}

n--; // Reduce size

if (j == -1); // check current index again after shift

{
 printf("Value not found in the array\n");

 if (found) {

 printf("Array after deletion : ");

 for (i = 0; i < n; i++) {

 printf("%d ", array[i]);

 printf("\n");

}

 return 0;

}

Output:-

Enter the number of elements in the array: 5

Enter 5 elements: 1 2 3 5 7

Enter the value to delete: 3

Array after deletion: 1 2 5 7

19 Search a value within an array:

= (Total of code is not needed for this question)

⇒ #include <stdio.h>

int main() {

 int array[100], n, i, value, found = 0;

 printf("Enter the number of elements in
the array: ");

```

scanf("%d", &n);
printf("Enter %d elements: \n", n);
for (i=0; i<n; i++) {
    scanf("%d", &arr[i]);
}
printf("Enter the value to search: ");
scanf("%d", &value);
for (i=0; i<n; i++) {
    if (arr[i] == value) {
        printf("Value %d found at position
                %d \n", value, i+1);
        found = 1;
        break;
    }
}
if (!found) {
    printf("Value %d not found in the
            array. \n", value);
}
return 0;

```

Output: Enter the number of elements = 5
 Enter 5 elements: 1 2 3 5 7
 Enter the value to search: 5
 Value 5 found at position: 7

Section B: Programs Using Strings

1. Find length of a string.

=> #include <stdio.h>

#include <string.h>

int main()

char str[700];

int length;

printf ("Enter a string:");

gets(str); // or use fgets(str,
sizeof(str), stdin);

length = strlen(str);

printf ("Length of the string = :

%d\n", length);

return 0;

Output: Enter the string : Hello

The length of the string is: 5

2. Convert a string to lowercase.

=> #include <stdio.h>

#include <ctype.h>

int main()

```

char str[100];
int i;

printf ("Enter a string : ");
gets(str); // or use fgets(str, 100, stdin);

for(i=0; str[i] != '\0'; i++) {
    str[i] = tolower(str[i]);
}

printf ("Lowercase string : %s", str);
return 0;
}

```

Output:- Enter a string : HELLO WORLD
 Lowercase string : hello world

3 Convert a string to uppercase:

```

#include <stdio.h>
#include <ctype.h>

int main() {
    char str[100];
    int i;

    printf ("Enter a string : ");
    gets(str); // or use fgets(str, 100, stdin);

    for(i=0; str[i] != '\0'; i++) {
        str[i] = toupper(str[i]);
    }
}

```

```
    printf ("Uppercase string : %.s", str);
    return 0;
}
```

Output: Enter a string : hello world
Uppercase string: HELLO WORLD

4 Convert a string to toggle case. (eg. ABC → aBc)

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

```
#include <ctype.h>
```

```
int main() {
```

```
    char str[100];
```

```
    int i;
```

```
    printf ("Entered a string: ");
```

```
    gets(str); // or use fgets(str, 100, stdin);
```

```
    for (i = 0; str[i] != '\0'; i++) {
```

```
        if (islower(str[i]))
```

```
            str[i] = toupper(str[i]);
```

```
        else if (isupper(str[i]))
```

```
            str[i] = tolower(str[i]);
```

```
}
```

```
    printf ("Toggled string: %.s", str);
```

```
    return 0;
}
```

Output: Enter a string : AbCd

Toggled string : aBcD

5 Copy one string to another

```
#include <stdio.h>
int main() {
    char str1[100], str2[100];
    int i;
    printf("Enter a string:");
    gets(str1); // or use fgets(str1, 100, stdin);
    for(i=0; str1[i] != '\0'; i++) {
        str2[i] = str1[i];
    }
    str2[i] = '\0'; // End the copied string properly
    printf("Copied string: %s", str2);
    return 0;
}
```

Output: Enter a string: Hello World
Copied string: Hello World

6 Compare two strings lexicographically and print which one is greater, smaller or same.

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

```
int main() {
```

```
char str1[100], str2[100];
```

```
int result;
```

```
printf("Enter first second string: ");  
gets(str1); // or use fgets(str1, 100, stdin);  
printf("Enter second string: ");  
gets(str2); // or use fgets(str2, 100, stdin);
```

```
result = strcmp(str1, str2);
```

```
if (result == 0)
```

```
    printf("Both strings are SAME.");
```

```
else if (result < 0)
```

```
    printf("First string is SMALLER.");
```

```
else
```

```
    printf("First string is GREATER.");
```

```
return 0;
```

Output: Enter First string: Apple

Enter second string: Banana

First string is SMALLER.

7

Reverse a string

=) #include <stdio.h>

#include <string.h>

```
int main()
```

```
char str[100]; rev[100];
```

~~char temp;~~

```
int i, len;
```

```
printf ("Enter a string: ");
gets(str); // or use fgets(str, 100, stdin);
```

```
len = strlen(str);
```

```
for (i = 0; i < len; i++) {
```

```
    rev[i] = str[len - i - 1];
```

```
rev[len] = '\0'; // end of string
```

```
printf ("Reversed string: %s", rev);
```

```
return 0;
```

```
}
```

Output: Enter a string: 72

Reversed string: 27

Q Check whether a string is a palindrome.

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

```
#include <string.h>
```

```
int main () {
```

```
char str [100];
```

```
int i, len, flag = 0;
```

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

```
gets(str); // or use fgets(str, 100, stdin);
```

```
len = strlen(str);
```

```

for(i=0; i < len/2; i++) {
    if(str[i] != str[len-i-1]) {
        flag = 1;
        break;
    }
}
if(flag == 0)
    printf ("The string is a palindrome.");
else
    printf ("The string is NOT a Palindrome.");
return 0;
}

```

Output: Enter a string: hello
 The string is NOT a Palindrome.

Enter a string: Madam
 The string is a Palindrome.

9) Concatenate one string at the end of another string

```

#include <stdio.h>
#include <string.h>

int main() {
    char str1[100], str2[50];

```

```

    printf ("Enter the first string:");
    fgets (str1, 100, stdin);

```

```
// str1 [strcspn(str1, "\n")] = '\0'; //  
remove newline
```

```
printf("Enter the second string: ");  
fgets(str2, 50, stdin);  
if(str2[strcspn(str2, "\n")] != '\0'); //  
remove newline  
strcat(str1, str2); // Concatenate str2 at the end of str1
```

```
printf("After concatenation: %.5s\n",  
str1);  
return 0;
```

Output: Enter the first string : Hello
 Enter the second string : world
 After concatenation: Hello World

7.0 Point characters of a string vertically

=> #include <stdio.h>

#include <string.h>

```
int main() {  
    char str[100];  
    int i, len;  
  
    printf("Enter a string: ");  
    fgets(str, 100, stdin);
```

: 11. `strlstrip(str, "ln") == 'lo';` //
remove newline

```
if (len != strlen(str)) {
    printf("characters vertically: \n");
    for (i = 0; i < len; i++) {
        printf("%c\n", str[i]);
    }
}
```

Output: Enter a string: Hello
characters vertically:

H

e

l

o

11. Print reversed string vertically character by character.

=> #include <stdio.h>
#include <string.h>

```
int main() {
    char str[100];
    int i, len;
```

printf("Enter a string: ");

```

fgets(str, 100, stdin);
if [str[spn] == '\n'] = '0' ; // remove newline
len = strlen(str);
printf("Reversed string vertically:\n");
for(i = len - 1; i >= 0; i--) {
    printf("%c\n", str[i]);
}
return 0;

```

Output: Enter a string ::Hello

Reversed string vertically:

O
I
E
H

Q2 Print frequency of each vowel in given string.

```

=> #include <stdio.h>
#include <string.h>
#include <ctype.h>

```

```

int main() {
    char str[100];
    int i;

```

```

int a_count = 0, e_count = 0, i_count = 0,
o_count = 0, u_count = 0;
printf ("Enter a string : ");
fgets (str, 100, stdin);
str [strcspn (str, "\n")] = '\0'; // remove newline
for (i = 0; str[i] != '\0'; i++) {
    char ch = tolower (str[i]); // convert to lowercase for uniformity
    switch (ch) {
        case 'a': a_count++; break;
        case 'e': e_count++; break;
        case 'i': i_count++; break;
        case 'o': o_count++; break;
        case 'u': u_count++; break;
    }
}

```

```

printf ("Frequency of vowels:\n");
printf ("a: %.d\n", a_count);
printf ("e: %.d\n", e_count);
printf ("i: %.d\n", i_count);
printf ("o: %.d\n", o_count);
printf ("u: %.d\n", u_count);

```

return 0;

3.

Output : Enter a string : Education
Frequency of vowels:

a : 2

e : 2

i : 1

o : 1

u : 1