TOPIC NO.: 1 TITLE: Implementation of 1D Array DATE: 17.01.24

ALGORITHM:

Step-1: Start

Step-2: Read r

Step-3: Declare a 1-D array "arr"with length r

Step-4: Repeat for i=0 to r-1

taking input from user in the array

[END OF LOOP]

Step-5: Repeat for i=0 to r-1

print the elements of array

[END OF LOOP]

Step-6: End

PROGRAM CODE:

#include <stdio.h>

int main()

{

int row;

printf("Write array size: ");

scanf("%d", &row);

int arr[row];

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

{

printf("%d = ", i);

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

}

printf("Array = \n");

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

{

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

}

return 0;

}

OUTPUT:

Write the length of array: 5

0=12

1=21

2=3

3=43

4 = 56

Array: 12 21 3 43 56

TOPIC NO.: 2 TITLE: Selection-sorting of 1D Array DATE: 17.01.24

ALGORITHM:

Step-1: Start

Step-2: Read r

Step-3: Declare a 1-D array 'arr' with length r

Step-4: Input elements using for loop

Step-5: Repeat for i=0 to r-1

for j=i+1 to r-1

IF arr[i] > arr[j]

SET temp = arr[i]

SET arr[i] = arr[j]

SET arr[j] = temp

[END OF IF]

[END OF LOOP]

Step-6: Print the sorted array using for loop

Step-7: End

PROGRAM CODE:

#include <stdio.h>

int main() {

int i, r, j, t;

printf("Enter the length of array: ");

scanf("%d", &r);

int arr[r];

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

printf("element at %d = ", i);

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

}

for (i = 0; i < r; i++)

{

for (j = i + 1; j < r; j++){

if (arr[i] > arr[j]) {

t = arr[i];

arr[i] = arr[j];

arr[j] = t;

}

}

}

printf("Sorted Array: ");

for (j = 0; j < r; j++)

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

return 0;

}

OUTPUT:

Enter the length of array: 4

0=12

1=21

2=3

3=43

Sorted Array: 3 12 21 43

TOPIC NO.: 3 TITLE: Implementation of 2D Array DATE: 24.01.24

ALGORITHM:

Step-1: Start

Step-2: Read r and c

Step-3: Declare a 2-D array 'arr' with dimension r x c

Step-4: Repeat

for i=0 to r-1

for j=0 to c-1

taking input from user in the array

[END OF LOOP]

Step-5: Repeat

for i=0 to r-1

for j=0 to c-1

print the elements of the array

[END OF LOOP]

Step-6: End

PROGRAM CODE:

#include <stdio.h>

int main() {

int r, c;

printf("Enter the number of rows and columns: ");

scanf("%d%d", &r, &c);

int arr[r][c];

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

for (int j = 0; j < c; j++) {

printf("Enter element at [%d][%d]: ", i, j);

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

} }

printf("Array:\n");

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

for (int j = 0; j < c; j++) {

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

}

printf("\n");

}

return 0;

}

OUTPUT:

Enter the number of rows and columns: 33

Enter element at [0][0]: 1

Enter element at [0][1]: 2

Enter element at [0][2]: 3

Enter element at [1][0]: 4

Enter element at [1][1]: 5

Enter element at [1][2]: 6

Enter element at [2][0]: 7

Enter element at [2][1]: 8

Enter element at [2][2]: 9

Array:

1 2 3

4 5 6

7 8 9

TOPIC NO.: 4 TITLE: Selection-sorting of 2-D Array DATE: 24.01.24

ALGORITHM:

Step-1: Start

Step-2: Read r and c

Step-3: Declare a 2-D array 'arr' with dimension r x c

Step-4: Input elements using nested for loops

Step-4: Repeat

for i=0 to r-l

for j=0 to c-1

for k=0 to r-1

for l=0 to c-1

IF arr[i][j]<arr[k][I]

SET temp = arr[i][j]

SET arr[i][j]=arr[k][l]

SET arr[i][j]=temp

[END OF IF]

[END OF LOOP]

Step-5: Print the sorted array using nested for loops

Step-6: End

PROGRAM CODE:

#include <stdio.h>

int main()

{

int i, j, r, c, m, n, t;

printf("Enter the number of rows and columns: ");

scanf("%d %d", &r, &c);

int arr[r][c];

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

for (j = 0; j < c; j++){

printf("Enter element at [%d][%d]: ", i, j);

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

}

}

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

for (j = 0; j < c; j++) {

for (m = 0; m < r; m++) {

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

if (arr[i][j] < arr[m][n]) {

t = arr[i][j];

arr[i][j] = arr[m][n];

arr[m][n] = t;

}

}

}

}

}

for (m = 0; m < r; m++) {

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

printf("%d ", arr[m][n]);

}

printf("\n");

}

return 0;

}

OUTPUT:

Enter the no. of rows and columns: 3

3

Enter element at [0][0]: 12

Enter element at [0][1]: 34

Enter element at [0][2]: 2

Enter element at [1][0]: 13

Enter element at [1][1]: 65

Enter element at [1][2]: 4

Enter element at [2][0]: 56

Enter element at [2][1]: 78

Enter element at [2][2]: 4

Sorted Array:

2 4 4

12 13 34

56 65 78