**DS Assignment - Familiarization of Arrays**

-Aldrin Jenson

CS3B

Roll No.5

**1. Write a program in C to merge two arrays of the same size, sorted in descending order.**

#include <stdio.h>

void main()

{

int len = 4, i, j, k;

int a[4] = {14, 6, 5, 2};

int b[4] = {10, 9, 5, 1};

int sorted[10], max;

for (i = 0, j = 0, k = 0; i < len && j < len; k++)

{

if (a[i] > b[j])

{

max = a[i];

i++;

}

else

{

max = b[j];

j++;

}

sorted[k] = max;

}

for (; i < len; i++, k++)

sorted[k] = a[i];

for (; j < len; k++, j++)

sorted[k] = b[j];

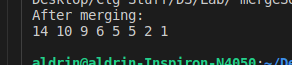
printf("After merging:\n");

for (int i = 0; i < len \* 2; i++)

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

}

**O/P:**



**2. Write a program in C to accept a matrix and determine whether it is a sparse matrix.**

#include <stdio.h>

void main()

{

int r, c, a[5][5], count = 0;

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

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

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

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

{

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

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

if (a[i][j] == 0)

count++;

}

if (count > (r \* c) / 2)

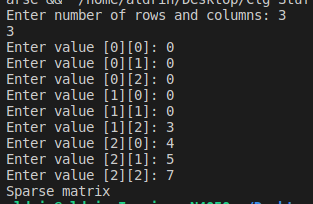
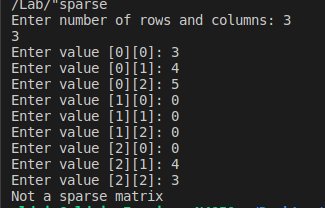
printf("Sparse matrix");

else

printf("Not a sparse matrix");

}

O/P:



**3. Write a program in C to rotate an array by N positions**

#include <stdio.h>

void main()

{

int a[20], size, n, i, temp;

printf("Enter array size: ");

scanf("%d", &size);

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

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

printf("Enter roating amount: ");

scanf("%d", &n);

printf("\nEntered array: ");

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

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

for (i = 0; i < size - n; i++)

{

temp = a[i];

a[i] = a[i + n];

a[i + n] = temp;

}

for (; i < size - 1; i++)

{

temp = a[i];

a[i] = a[i + 1];

a[i + 1] = temp;

}

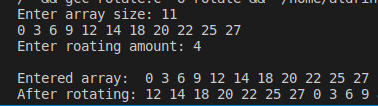
printf("\nAfter rotating: ");

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

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

}

O/P:



**4. Write a program in C to print a matrix in spiral form**

#include <stdio.h>

void main()

{

int r = 6, c = 5;

int a[6][5] = {{1, 2, 3, 4, 5},

{6, 7, 8, 9, 10},

{11, 12, 13, 14, 15},

{16, 17, 18, 19, 20},

{21, 22, 23, 24, 25},

{26, 27, 28, 29, 30}};

int ans[30];

int s = r \* c, k;

int i, j;

int iVal = 0, jVal = 0, count = 0;

for (k = 0, i = 0, j = 0; k < s; k++)

{

ans[k] = a[i][j];

if (i == count + 1 && j == count)

{

count++;

}

if (i == count)

{

iVal = 0;

jVal = 1;

}

if (j == c - count - 1)

{

iVal = 1;

jVal = 0;

}

if (i == r - count - 1)

{

iVal = 0;

jVal = -1;

}

if (j == count && i != count)

{

iVal = -1;

jVal = 0;

}

i += iVal;

j += jVal;

}

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

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

}

O/P:



**5. Write a program in C to rearrange positive and negative numbers alternatively in a given array:**

#include <stdio.h>

void main()

{

int len = 11;

int i, j, k;

int a[11] = {-4, 8, -30, -6, 5, -9, 7, 1, -21, -11, 19};

int ans[11];

int positive[11], negative[11], positiveCount, negativeCount;

for (positiveCount = 0, negativeCount = 0, k = 0; k < len; k++)

{

if (a[k] > 0)

{

positive[positiveCount] = a[k];

positiveCount++;

}

else

{

negative[negativeCount] = a[k];

negativeCount++;

}

}

for (; positiveCount < len; positiveCount++)

positive[positiveCount] = 0;

for (; negativeCount < len; negativeCount++)

negative[negativeCount] = 0;

for (k = 0, i = 0, j = 0; k < len;)

{

if (positive[i])

{

ans[k] = positive[i];

i++;

k++;

}

if (negative[j])

{

ans[k] = negative[j];

j++;

k++;

}

}

printf("\nAfter rearranging: ");

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

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

}

O/P:

