1) #include <stdio.h>

void spiralPrint(int m, int n, int a[m][n]) {

int i, rowStart = 0, colStart = 0;

while (rowStart < m && colStart < n) {

for (i = colStart; i < n; ++i)

printf(”%d “, a[rowStart][i]);

rowStart++;

for (i = rowStart; i < m; ++i)

printf(”%d “, a[i][n - 1]);

n--;

if (rowStart < m) {

for (i = n - 1; i >= colStart; --i)

printf(”%d “, a[m - 1][i]);

m--;

}

if (colStart < n) {

for (i = m - 1; i >= rowStart; --i)

printf(”%d “, a[i][colStart]);

colStart++;

}

}

}

int main() {

int a[4][4] = {

{1, 2, 3, 4},

{5, 6, 7, 8},

{9, 10, 11, 12},

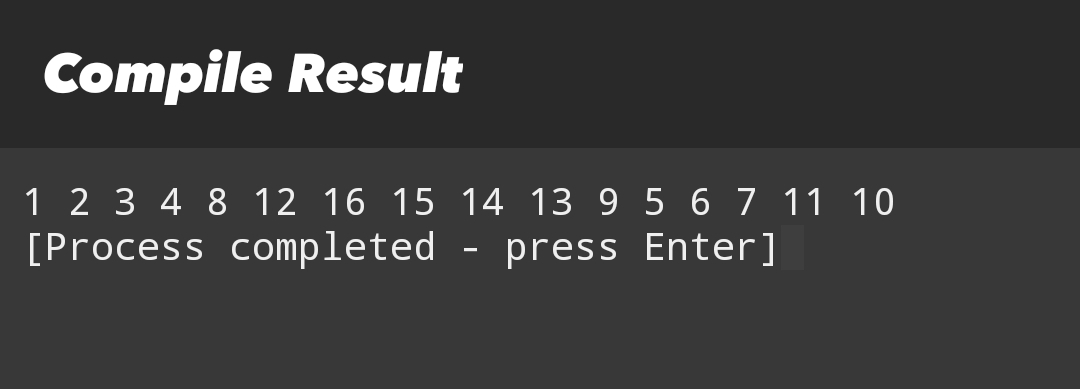
{13,14, 15, 16}

};

spiralPrint(4, 4, a);

return 0;

}



2) #include <stdio.h>

#define N 3

void rotate90(int a[N][N]) {

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

for (int j = i; j < N; j++) {

int temp = a[i][j];

a[i][j] = a[j][i];

a[j][i] = temp;

}

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

for (int j = 0; j < N / 2; j++) {

int temp = a[i][j];

a[i][j] = a[i][N - 1 - j];

a[i][N - 1 - j] = temp;

}

}

int main() {

int a[N][N] = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

rotate90(a);

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

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

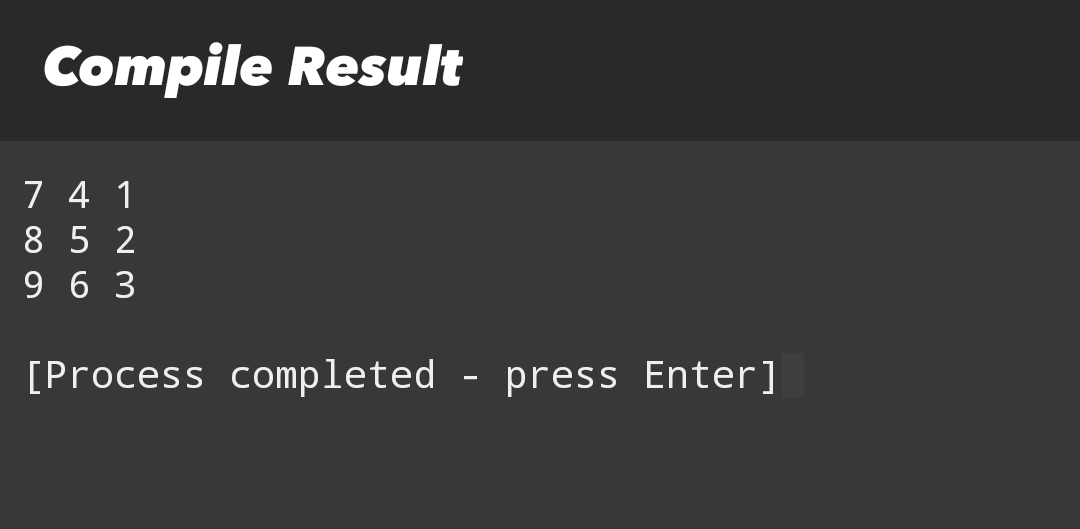
printf(”%d “, a[i][j]);

printf(”\n”);

}

return 0;

}



3) #include <stdio.h>

int sumDiagonals(int n, int a[n][n]) {

int sum = 0;

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

sum += a[i][i];

if (i != n - 1 - i)

sum += a[i][n - 1 - i];

}

return sum;

}

int main() {

int a[3][3] = {

{1, 2, 3},

{4, 5, 6},

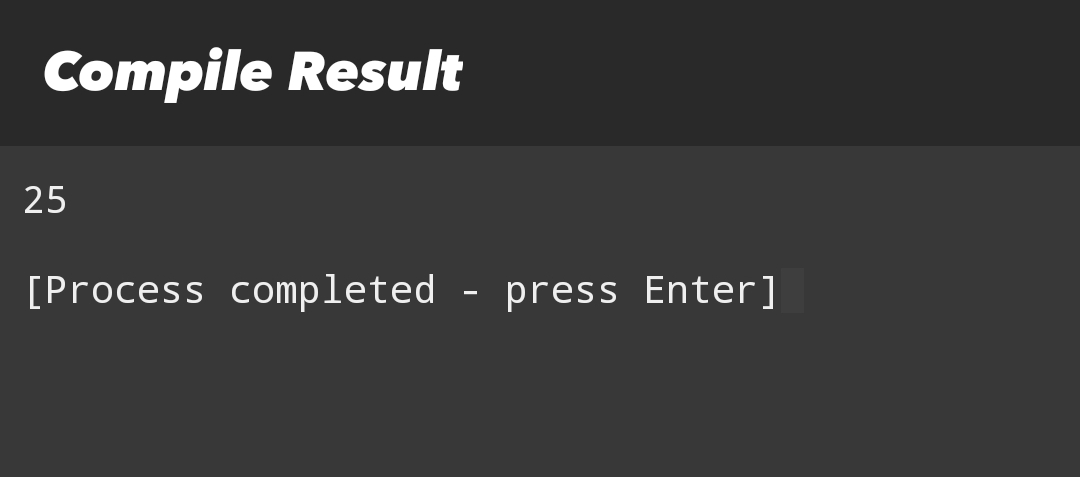
{7, 8, 9}

};

printf(”%d\n”, sumDiagonals(3, a));

return 0;

}



4) #include <stdio.h>

#define ROW 3

#define COL 3

int main() {

int a[ROW][COL] = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

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

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

printf(”%d “, a[j][i]);

printf(”\n”);

}

return 0;

}



5) #include <stdio.h>

int main() {

int a[3][3] = {

{0, 0, 1},

{0, 0, 0},

{4, 0, 0}

};

int rows = 3, cols = 3, zeroCount = 0;

int total = rows \* cols;

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

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

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

zeroCount++;

if (zeroCount > total / 2)

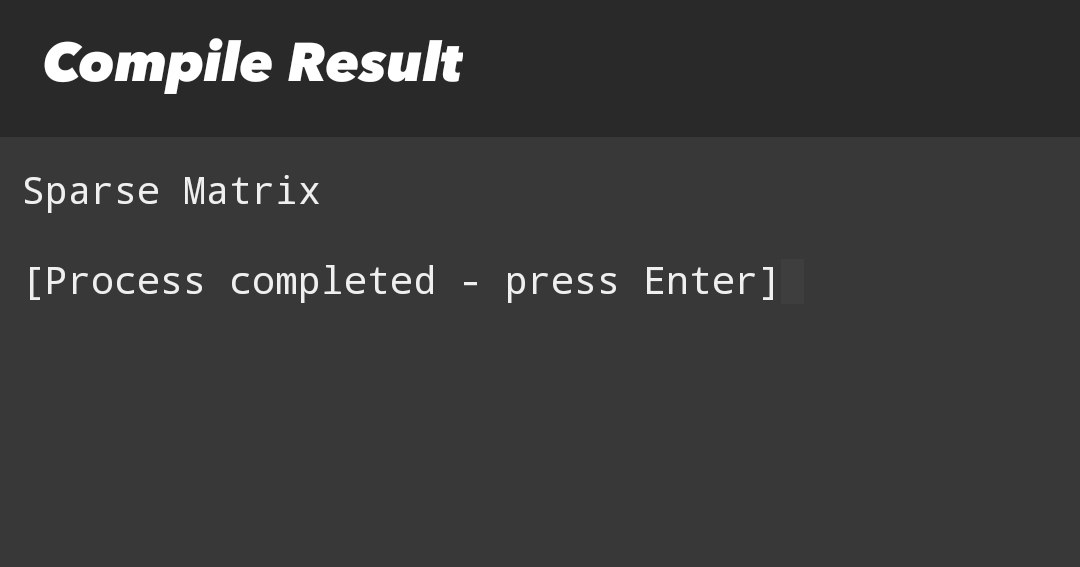
printf(”Sparse Matrix\n”);

else

printf(”Not Sparse\n”);

return 0;

}



6) #include <stdio.h>

#define SIZE 5

int main() {

int p1[SIZE] = {5, 0, 10, 6, 2};

int p2[SIZE] = {1, 2, 4, 1, 3};

int result[SIZE];

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

result[i] = p1[i] + p2[i];

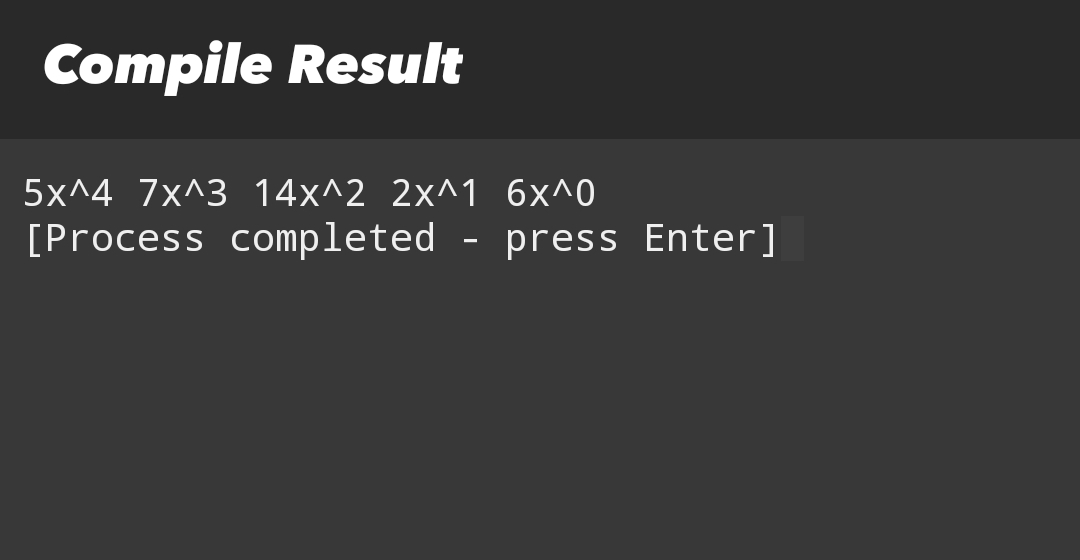
for (int i = SIZE - 1; i >= 0; i--)

if (result[i] != 0)

printf(”%dx^%d “, result[i], i);

return 0;

}



7) #include <stdio.h>

#define SIZE1 3

#define SIZE2 3

int main() {

int p1[SIZE1] = {1, 2, 3};

int p2[SIZE2] = {4, 5, 6};

int result[SIZE1 + SIZE2 - 1] = {0};

for (int i = 0; i < SIZE1; i++)

for (int j = 0; j < SIZE2; j++)

result[i + j] += p1[i] \* p2[j];

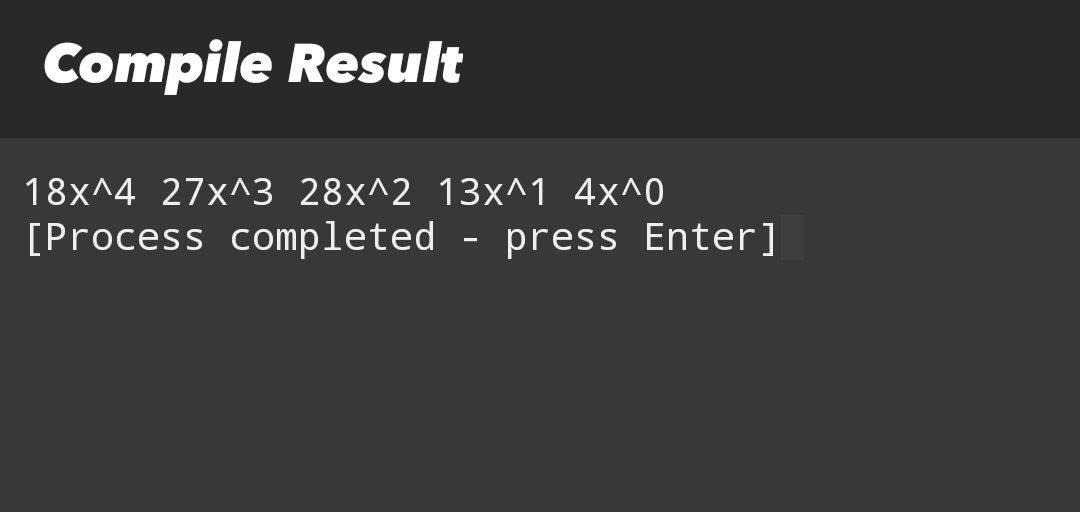
for (int i = SIZE1 + SIZE2 - 2; i >= 0; i--)

if (result[i] != 0)

printf(”%dx^%d “, result[i], i);

return 0;

}



8) #include <stdio.h>

#include <stdlib.h>

int main() {

int n = 2, q = 5;

int \*arr[n];

int sizes[n], lastAnswer = 0, type, x, y;

int query[5][3] = {

{1, 0, 5},

{1, 1, 7},

{1, 0, 3},

{2, 1, 0},

{2, 1, 1}

};

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

arr[i] = NULL;

sizes[i] = 0;

}

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

type = query[i][0];

x = query[i][1];

y = query[i][2];

int idx = (x ^ lastAnswer) % n;

if (type == 1) {

arr[idx] = realloc(arr[idx], (sizes[idx] + 1) \* sizeof(int));

arr[idx][sizes[idx]++] = y;

} else {

lastAnswer = arr[idx][y % sizes[idx]];

printf(”%d\n”, lastAnswer);

}

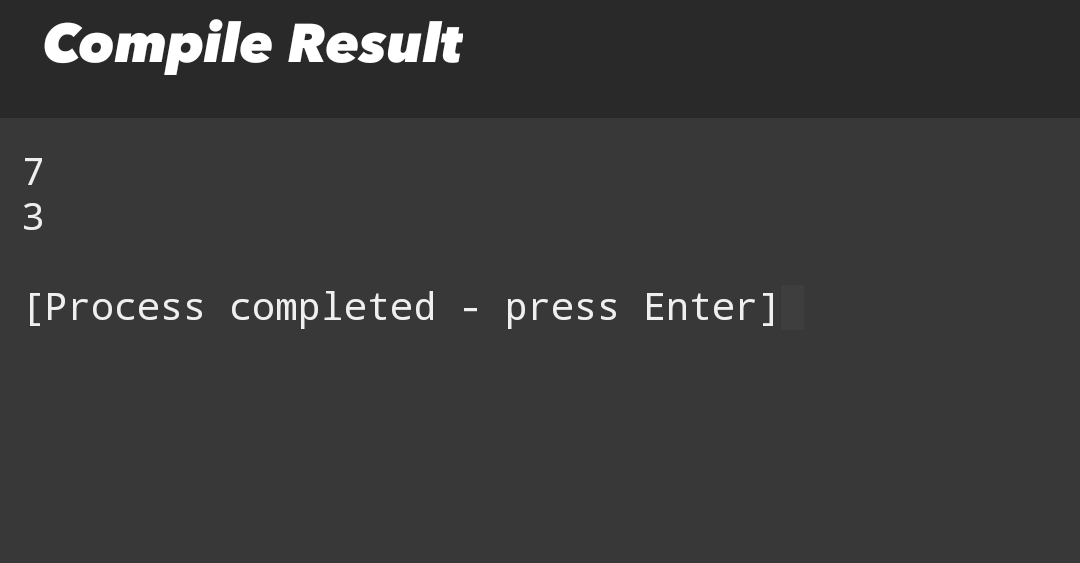
}

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

free(arr[i]);

return 0;

}



9) #include <stdio.h>

int main() {

int a[3][3] = {

{1, 0, 0},

{0, 1, 0},

{0, 0, 1}

};

int isIdentity = 1;

for (int i = 0; i < 3; i++)

for (int j = 0; j < 3; j++)

if ((i == j && a[i][j] != 1) || (i != j && a[i][j] != 0))

isIdentity = 0;

if (isIdentity)

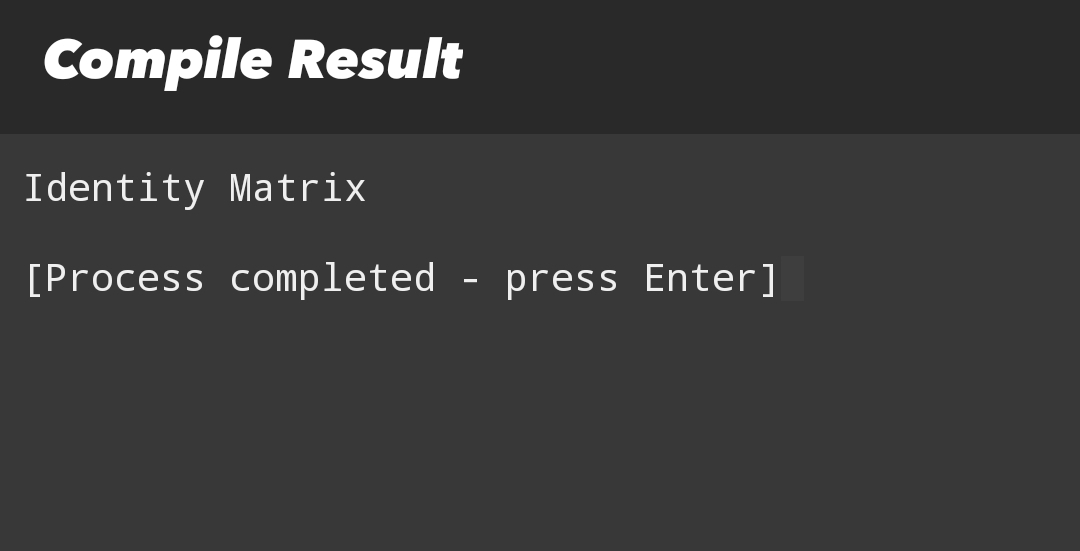
printf(”Identity Matrix\n”);

else

printf(”Not Identity Matrix\n”);

return 0;

}



10) #include <stdio.h>

int countZeros(int a[4][4], int n) {

int count = 0, i = 0, j = n - 1;

while (i < n && j >= 0) {

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

count += (j + 1);

i++;

} else {

j--;

}

}

return count;

}

int main() {

int a[4][4] = {

{0, 0, 1, 1},

{0, 1, 1, 1},

{0, 0, 0, 1},

{1, 1, 1, 1}

};

printf(”%d\n”, countZeros(a, 4));

return 0;

}

