/\*

Pointers \* -> pointer to a variable/ value\_at

int num = 10;

num

10

1000

address is always +ve integer

int \*ptr;

\*/

#include <stdio.h>

int main()

{

int num = 10;

int \*ptr;

ptr = &num;

printf("%d", \*(ptr)); // \*(1000)

return 0;

}

#include <stdio.h>

int main()

{

int num = 10; // num a b c

int \*a, \*\*b, \*\*\*c; // 10 1000 2000 4000

a = &num; // 1000 2000 4000 6000

b = &a;

c = &b;

printf("%d", \*\*\*c); // 10

return 0;

}

/\*

Pointer with array

int arr[7];

index - 0 1 2 3 4 5 6

val - 10 20 30 40 50 60 70

add - 1000 1004 1008 1012 1016 1020 1024

int \*ptr; arr + 1 = 1000 + 1 = 1004

ptr = arr; //1000 -> BASE ADDRESS ( 0th index add)

\*/

#include<stdio.h>

int main()

{

int arr[7]={10,20,30,40,50,60,70};

int \*ptr;

ptr = arr;

printf("%d", \*(ptr+2));

}

/\*

Pointer with array

// 30 60 80

//

\*/

#include<stdio.h>

int main()

{

int arr[10]={10,20,30,40,50,60,70,80,90,100};

int \*ptr1, \*ptr2, \*ptr3;

ptr1 = arr;

ptr2 = arr+2;

ptr3 = arr+3;

printf("%d %d %d", \*(ptr1+2), \*(ptr2+3), \*(ptr3+4));

}

/\*

Pointer with array

arr[ind] => \*(arr + ind);

\*/

#include<stdio.h>

void print(int \*arr) // arr[]

{

int ind;

for(ind = 0; ind < 10; ind++)

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

}

int main()

{

int arr[10]={10,20,30,40,50,60,70,80,90,100};

print(arr);

}

/\*

Pointer with array

arr[ind] => \*(arr + ind);

0 1 2 3 4 5 6 7

10 20 30 40 50 60 70 80

1000 1004 1008 1012

\*/

#include<stdio.h>

void print(int \*arr) // arr[]

{

int ind;

for(ind = 0; ind < 10; ind++)

arr[ind] = arr[ind] / 10;

for(ind = 0; ind < 10; ind++)

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

}

int main()

{

int arr[10]={10,20,30,40,50,60,70,80,90,100};

print(arr); // Call by reference

printf("\n");

for(int ind = 0; ind < 10; ind++)

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

}

/\*

Rotate Array in Right

10

10 20 30 40 50 60 70 80 90 100

4

70 80 90 100 10 20 30 40 50 60

\*/

#include<stdio.h>

void REVERSE( int arr[], int start, int end)

{

int f1 = start, f2 = end;

while( f1 < f2 )

{

int temp = arr[f1];

arr[f1++] = arr[f2];

arr[f2--] = temp;

}

}

int main()

{

int SIZE, NOR, ind;

scanf("%d", &SIZE);

int arr[SIZE];

for(ind = 0; ind < SIZE; ind++)

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

scanf("%d", &NOR);

NOR = NOR % SIZE;

if( NOR != 0 )

{

REVERSE( arr, 0, SIZE-1 );

REVERSE( arr, 0, NOR-1 );

REVERSE( arr, NOR, SIZE-1 );

}

for(ind = 0; ind < SIZE; ind++)

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

}

/\*

Rotate Array in Left

10

10 20 30 40 50 60 70 80 90 100

4

50 60 70 80 90 100 10 20 30 40

\*/

#include<stdio.h>

void REVERSE( int\*, int , int);

void REVERSE( int \*arr, int start, int end)

{

int f1 = start, f2 = end;

while( f1 < f2 )

{

int temp = arr[f1];

arr[f1++] = arr[f2];

arr[f2--] = temp;

}

}

int main()

{

int SIZE, NOR, ind;

scanf("%d", &SIZE);

int arr[SIZE];

for(ind = 0; ind < SIZE; ind++)

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

scanf("%d", &NOR);

NOR = NOR % SIZE;

if( NOR != 0 )

{

REVERSE( arr, 0, NOR-1 );

REVERSE( arr, NOR, SIZE-1 );

REVERSE( arr, 0, SIZE-1 );

}

for(ind = 0; ind < SIZE; ind++)

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

}

/\*

Find the row which has highest number of Zeros.

if both rows are in same count just give the higher index.

4 4

1 0 1 0

2 0 0 5

0 1 0 0

4 2 6 5

2

\*/

#include<stdio.h>

int main()

{

int NROW, NCOL, row, col, count, max, max\_ind;

scanf("%d %d", &NROW, &NCOL);

int arr[NROW][NCOL];

for(row = 0; row < NROW; row++)

for(col = 0; col < NCOL; col++)

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

max = 0;

max\_ind = 0;

for(row = 0; row < NROW; row++)

{

for(col = 0, count = 0; col < NCOL; col++)

{

if(arr[row][col] == 0) count++;

}

if( count >= max )

{

max = count;

max\_ind = row;

}

}

if(max == 0) printf("No Zeros in the array");

else printf("%d", max\_ind);

}