Qno.1) #include <stdio.h>

int main() {

int size, i, j, k, temp, m;

// Read in the array size

printf("Enter an array size: ");

scanf("%d", &size);

// Allocate memory for the array

int arr[size];

// Read in the array elements

printf("Enter numbers for your array: ");

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

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

}

// Read in the number of rotations

printf("Enter number of rotations: ");

scanf("%d", &m);

// Perform m rotations

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

// Save the first element

temp = arr[0];

// Shift the remaining elements to the left

for (j = 0; j < size - 1; j++) {

arr[j] = arr[j+1];

}

// Put the first element at the end

arr[size-1] = temp;

}

// Print the rotated array

printf("Results: ");

for (k = 0; k < size; k++) {

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

}

printf("\n");

return 0;

}

Qno.2)

#include <stdio.h>

int main() {

int road\_length, num\_areas, start, end, trees\_to\_move = 0;

// get input from user

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

scanf("%d", &road\_length);

printf("Enter how many areas are needed for new facilities: ");

scanf("%d", &num\_areas);

// initialize an array to keep track of tree locations

int trees[road\_length+1];

for (int i = 0; i <= road\_length; i++) {

trees[i] = 1;

}

// mark the trees in the areas to be cleared

for (int i = 1; i <= num\_areas; i++) {

printf("area#%d starting point & ending point: ", i);

scanf("%d %d", &start, &end);

for (int j = start; j <= end; j++) {

trees[j] = 0;

}

}

// count the number of trees to move

for (int i = 0; i <= road\_length; i++) {

if (trees[i] == 0) {

trees\_to\_move++;

}

}

// print the result

printf("Result of number of trees needed to move: %d", trees\_to\_move);

return 0;

}

Qno.3)

#include <stdio.h>

int main() {

int num;

printf("Enter a number: ");

scanf("%d", &num);

if (num % 3 == 0 && num % 5 == 0 && num % 7 == 0) {

printf("%d is divisible by 3, 5, and 7\n", num);

} else if (num % 3 == 0 && num % 5 == 0) {

printf("%d is divisible by 3 and 5\n", num);

} else if (num % 3 == 0 && num % 7 == 0) {

printf("%d is divisible by 3 and 7\n", num);

} else if (num % 5 == 0 && num % 7 == 0) {

printf("%d is divisible by 5 and 7\n", num);

} else if (num % 3 == 0) {

printf("%d is divisible by 3\n", num);

} else if (num % 5 == 0) {

printf("%d is divisible by 5\n", num);

} else if (num % 7 == 0) {

printf("%d is divisible by 7\n", num);

} else {

printf("NOT\n");

}

return 0;

}

Qno.4)

#include <stdio.h>

int main() {

char student;

char largest\_lake, smallest\_lake, second\_biggest\_lake, third\_biggest\_lake;

// get input from user for which student's description to use

printf("Enter which student's description to use (a, b, c, or d): ");

scanf(" %c", &student);

// determine which lake is largest, smallest, second biggest, and third biggest based on the student's description

switch (student) {

case 'a':

largest\_lake = 'd';

smallest\_lake = 'h';

second\_biggest\_lake = 'b';

third\_biggest\_lake = 't';

break;

case 'b':

largest\_lake = 'h';

smallest\_lake = 'd';

second\_biggest\_lake = 'b';

third\_biggest\_lake = 't';

break;

case 'c':

largest\_lake = 'd';

smallest\_lake = 'h';

second\_biggest\_lake = 't';

third\_biggest\_lake = 'b';

break;

case 'd':

largest\_lake = 'b';

smallest\_lake = 't';

second\_biggest\_lake = 'h';

third\_biggest\_lake = 'd';

break;

default:

printf("Invalid input\n");

return 0;

}

// print the size sequence of the lakes from big to small

printf("%c > %c > %c > %c\n", largest\_lake, second\_biggest\_lake, third\_biggest\_lake, smallest\_lake);

return 0;

}

Qno.5)

#include <stdio.h>

int main() {

int n;

printf("Enter an integer: ");

scanf("%d", &n);

printf("%d ", n);

while (n != 1) {

if (n % 2 == 0) {

n /= 2;

} else {

n = 3 \* n + 1;

}

printf("%d ", n);

}

return 0;

}