NAME- ANIMESH ANAND

Reg no.-22MIS0549

Subject-Data Structure and Algorithms

Question 1--

```
#include <stdio.h>
#include <stdbool.h>
#include <math.h>

bool is_prime(int n) {
    if (n <= 1) {
        return false;
    }
    for (int i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}

int main() {
    int A[] = {7, 21, 18, 3, 12};
    int n = sizeof(A) / sizeof(A[0]);
    int queue[n], stack[n];
    int queue_size = 0, stack_size = 0;
    for (int i = 0; i < n; i++) {
        if (is_prime(A[i])) {
            queue[queue_size++] = A[i];
        } else {
            stack[stack_size++] = A[i];
        }
    }

    printf("Queue: ");
    for (int i = 0; i < queue_size; i++) {
        printf("%d ", queue[i]);
    }
    printf("\n");

    printf("Stack: ");
    for (int i = stack_size - 1; i >= 0; i--) {
        printf("%d ", stack[i]);
    }
    printf("\n");

    return 0;
}
```

Question 2--

```
#include <stdio.h>
```

```
#include <stdlib.h>
#define MAX_SIZE 100

int stack1[MAX_SIZE], stack2[MAX_SIZE];
int top1 = -1, top2 = -1;

void enqueue(int x) {
    if (top1 == MAX_SIZE - 1) {
        printf("Queue is full\n");
        return;
    }
    stack1[++top1] = x;
}

int dequeue() {
    if (top1 == -1 && top2 == -1) {
        printf("Queue is empty\n");
        return -1;
    }
    if (top2 == -1) {
        while (top1 != -1) {
            stack2[++top2] = stack1[top1--];
        }
    }
    int x = stack2[top2--];
    return x;
}

int main() {
    int A[] = {1, 2, 3, 4, 5};
    int n = sizeof(A) / sizeof(A[0]);

    for (int i = 0; i < n; i++) {
        enqueue(A[i]);
    }

    printf("Queue contents: ");
    for (int i = 0; i < n; i++) {
        int x = dequeue();
        printf("%d ", x);
    }
    printf("\n");

    return 0;
}
```

Question 3—

```
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number of buildings: ");
    scanf("%d", &n);

    int heights[n];
    printf("Enter the heights of the buildings: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &heights[i]);
    }

    int stamina = heights[0], max_height = heights[0];
    for (int i = 1; i < n; i++) {
        if (heights[i] > max_height) {
            stamina ^= heights[i];
            max_height = heights[i];
        }
    }

    printf("Stamina required for the journey: %d\n", stamina);

    return 0;
}
```