Rajalakshmi Engineering College

Name: eswar rs
Email: 240801074@rajalakshmi.edu.in
Roll no: 240801074
Phone: 7845648127
Branch: REC
Department: I ECE AE
Batch: 2028
Degree: B.E - ECE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 4_CY

Attempt : 1
Total Mark : 30
Marks Obtained : 30

Section 1 : Coding

1. Problem Statement

Saran is developing a simulation for a theme park where people wait in a queue for a popular ride.

Each person has a unique ticket number, and he needs to manage the queue using a linked list implementation.

Your task is to write a program for Saran that reads the number of people in the queue and their respective ticket numbers, enqueue them, and then calculate the sum of all ticket numbers to determine the total ticket value present in the queue.

Input Format

The first line of input consists of an integer N, representing the number of people in the queue.

The second line consists of N space-separated integers, representing the ticket numbers.

Output Format

The output prints an integer representing the sum of all ticket numbers.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
2 4 6 7 5

Output: 24

Answer

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

struct Node {
    int data;
    struct Node* next;
};

struct Queue {
    struct Node* front;
    struct Node* rear;
};

void initQueue(struct Queue* q) {
    q->front = q->rear = NULL;
}

void enqueue(struct Queue* q, int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;

    if (q->rear == NULL) {
        q->front = q->rear = newNode;
    } else {
        q->rear->next = newNode;
        q->rear = newNode;
    }
}

int calculateSum(struct Queue* q) {
    int sum = 0;
    struct Node* temp = q->front;

    while (temp != NULL) {
        sum += temp->data;
        temp = temp->next;
    }

    return sum;
}

int main() {
    int n, i, value;
    struct Queue q;
    initQueue(&q);

    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        scanf("%d", &value);
        enqueue(&q, value);
    }

    int total = calculateSum(&q);
    printf("%d\n", total);

    return 0;
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Sara builds a linked list-based queue and wants to dequeue and display all positive even numbers in the queue. The numbers are added at the end of the queue.

Help her by writing a program for the same.

Input Format

The first line of input consists of an integer N, representing the number of elements Sara wants to add to the queue.

The second line consists of N space-separated integers, each representing an element to be enqueued.

Output Format

The output prints space-separated the positive even integers from the queue, maintaining the order in which they were enqueued.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
1 2 3 4 5

Output: 2 4

Answer

```
// You are using GCC
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* next;
};

struct Queue {
   struct Node* front;
   struct Node* rear;
```

```
};

void initQueue(struct Queue* q) {
    q->front = q->rear = NULL;
}

void enqueue(struct Queue* q, int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;

    if (q->rear == NULL) {
        q->front = q->rear = newNode;
    } else {
        q->rear->next = newNode;
        q->rear = newNode;
    }
}

int dequeue(struct Queue* q) {
    if (q->front == NULL)
        return -1;

    struct Node* temp = q->front;
    int val = temp->data;
    q->front = q->front->next;

    if (q->front == NULL)
        q->rear = NULL;

    free(temp);
    return val;
}

int isEmpty(struct Queue* q) {
    return q->front == NULL;
}

int main() {
    int n, i, value;
    struct Queue q;
    initQueue(&q);

    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        scanf("%d", &value);
        enqueue(&q, value);
    }

    while (!isEmpty(&q)) {
        int val = dequeue(&q);
        if (val > 0 && val % 2 == 0) {
            printf("%d ", val);
        }
    }

    return 0;
}
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

Pathirana is a medical lab specialist who is responsible for managing blood count data for a group of patients. The lab uses a queue-based system to track the blood cell count of each patient. The queue structure helps in processing the data in a first-in-first-out (FIFO) manner.

However, Pathirana needs to remove the blood cell count that is positive even numbers from the queue using array implementation of queue, as they are not relevant to the specific analysis he is performing. The remaining data will then be used for further medical evaluations and reporting.

Input Format

The first line consists of an integer n, representing the number of a patient's blood cell count.

The second line consists of n space-separated integers, representing a blood cell count value.

Output Format

The output displays space-separated integers, representing the remaining blood cell count after removing the positive even numbers.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
1 2 3 4 5

Output: 1 3 5

Answer

```
#include <stdio.h>

int main() {
    int n;
    scanf("%d", &n);

    int queue[15];
    int result[15];
    int i, front = 0, rear = 0, res_index = 0;

    for (i = 0; i < n; i++) {
        scanf("%d", &queue[rear]);
        rear++;
    }

    while (front < rear) {
        int val = queue[front];
        front++;

        if (!(val > 0 && val % 2 == 0)) {
            result[res_index++] = val;
        }
    }

    for (i = 0; i < res_index; i++) {
        printf("%d ", result[i]);
    }

    return 0;
}
```

Status : Correct *Marks : 10/10*