**Assignment 4**

**Question 1:**

#include <stdio.h>

#define SIZE 5

int queue[SIZE], front = -1, rear = -1;

void enqueue(int x) {

if (rear == SIZE - 1) printf("Queue is full\n");

else {

if (front == -1) front = 0;

queue[++rear] = x;

}

}

void dequeue() {

if (front == -1 || front > rear) printf("Queue is empty\n");

else printf("Dequeued: %d\n", queue[front++]);

}

void isEmpty() {

if (front == -1 || front > rear) printf("Queue is empty\n");

else printf("Queue is not empty\n");

}

void isFull() {

if (rear == SIZE - 1) printf("Queue is full\n");

else printf("Queue is not full\n");

}

void display() {

if (front == -1 || front > rear) printf("Queue is empty\n");

else {

printf("Queue: ");

for (int i = front; i <= rear; i++) printf("%d ", queue[i]);

printf("\n");

}

}

void peek() {

if (front == -1 || front > rear) printf("Queue is empty\n");

else printf("Front element: %d\n", queue[front]);

}

int main() {

int ch, x;

while (1) {

printf("\n1.Enqueue 2.Dequeue 3.isEmpty 4.isFull 5.Display 6.Peek 7.Exit\n");

scanf("%d", &ch);

switch (ch) {

case 1: scanf("%d", &x); enqueue(x); break;

case 2: dequeue(); break;

case 3: isEmpty(); break;

case 4: isFull(); break;

case 5: display(); break;

case 6: peek(); break;

case 7: return 0;

default: printf("Invalid choice\n");

}

}

}

**Question 2:**

#include <stdio.h>

#define SIZE 5

int cq[SIZE], front = -1, rear = -1;

void enqueue(int x) {

if ((front == 0 && rear == SIZE - 1) || (rear + 1) % SIZE == front) {

printf("Queue is full\n");

return;

}

if (front == -1) front = 0;

rear = (rear + 1) % SIZE;

cq[rear] = x;

}

void dequeue() {

if (front == -1) {

printf("Queue is empty\n");

return;

}

printf("Dequeued: %d\n", cq[front]);

if (front == rear) front = rear = -1;

else front = (front + 1) % SIZE;

}

void isEmpty() {

if (front == -1) printf("Queue is empty\n");

else printf("Queue is not empty\n");

}

void isFull() {

if ((front == 0 && rear == SIZE - 1) || (rear + 1) % SIZE == front)

printf("Queue is full\n");

else

printf("Queue is not full\n");

}

void display() {

if (front == -1) {

printf("Queue is empty\n");

return;

}

printf("Queue: ");

int i = front;

while (1) {

printf("%d ", cq[i]);

if (i == rear) break;

i = (i + 1) % SIZE;

}

printf("\n");

}

void peek() {

if (front == -1) printf("Queue is empty\n");

else printf("Front element: %d\n", cq[front]);

}

int main() {

int ch, x;

while (1) {

printf("\n1.Enqueue 2.Dequeue 3.isEmpty 4.isFull 5.Display 6.Peek 7.Exit\n");

scanf("%d", &ch);

switch (ch) {

case 1: scanf("%d", &x); enqueue(x); break;

case 2: dequeue(); break;

case 3: isEmpty(); break;

case 4: isFull(); break;

case 5: display(); break;

case 6: peek(); break;

case 7: return 0;

default: printf("Invalid choice\n");

}

}

}

**Question 3:**

#include <stdio.h>

#define SIZE 100

int main() {

int n;

printf("Enter even number of elements: ");

scanf("%d", &n);

int q[n];

printf("Enter elements:\n");

for (int i = 0; i < n; i++) scanf("%d", &q[i]);

int half = n / 2;

int res[n], k = 0;

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

res[k++] = q[i];

res[k++] = q[i + half];

}

printf("Interleaved queue: ");

for (int i = 0; i < n; i++) printf("%d ", res[i]);

return 0;

}

**Question 4:**

#include <stdio.h>

#include <string.h>

#define SIZE 100

int main() {

char s[SIZE];

printf("Enter string: ");

scanf("%s", s);

int freq[256] = {0};

char q[SIZE];

int front = 0, rear = -1;

for (int i = 0; i < strlen(s); i++) {

char c = s[i];

freq[c]++;

q[++rear] = c;

while (front <= rear && freq[q[front]] > 1) front++;

if (front > rear) printf("-1 ");

else printf("%c ", q[front]);

}

return 0;

}

**Question 5 (a):**

#include <stdio.h>

#define SIZE 100

int q1[SIZE], q2[SIZE];

int f1=0, r1=-1, f2=0, r2=-1;

void enqueue1(int x){ q1[++r1] = x; }

int dequeue1(){ return q1[f1++]; }

int empty1(){ return f1>r1; }

void enqueue2(int x){ q2[++r2] = x; }

int dequeue2(){ return q2[f2++]; }

int empty2(){ return f2>r2; }

void push(int x){

enqueue2(x);

while(!empty1()) enqueue2(dequeue1());

f1=0; r1=-1; f2=0;

while(f2<=r2) enqueue1(dequeue2());

r2=-1;

}

int pop(){

if(empty1()) return -1;

return dequeue1();

}

int main(){

push(10); push(20); push(30);

printf("Popped: %d\n", pop());

printf("Popped: %d\n", pop());

return 0;

}

**(b):**

#include <stdio.h>

#define SIZE 100

int q[SIZE];

int f=0, r=-1;

void enqueue(int x){ q[++r] = x; }

int dequeue(){ return q[f++]; }

int empty(){ return f>r; }

void push(int x){

enqueue(x);

int sz = r-f+1;

for(int i=0;i<sz-1;i++) enqueue(dequeue());

}

int pop(){

if(empty()) return -1;

return dequeue();

}

int main(){

push(5); push(15); push(25);

printf("Popped: %d\n", pop());

printf("Popped: %d\n", pop());

return 0;

}