**ASSIGNMENT-4**

Q1

#include <iostream>

using namespace std;

#define size 10

int queue[size];

int f=-1;

int r=-1;

int isfull(){

if(r==size-1){

return 1;

}

return 0;

}

int isempty(){

if(f==r){

return 1;

}

return 0;

}

void enqueue(int val){

if(isfull()){

cout<<"queue is full"<<endl;

}

else {

r++;

queue[r]=val;

}

}

void dequeue(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

f++;

cout<<queue[f]<<"dequeued"<<endl;

}

}

void display(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

for(int i=f+1;i<size;i++){

cout<<queue[i]<<endl;

}

}

}

void peek(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

cout<<"front most element is:"<<queue[f+1];

}

}

int main()

{

int value;

int choice;

while(true){

cout << "\n-- Queue Menu --\n";

cout << "1. Enqueue \n2.Dequeue \n3. isempty\n4. isfull\n5. Peek\n6. Display\n";

cout<<"enter your choice:"<<endl;

cin>>choice;

switch(choice){

case 1:

cout<<"enter value you want to enqueue:"<<endl;

cin>>value;

enqueue(value);

break;

case 2:

dequeue();

break;

case 3:

isempty();

break;

case 4:

isfull();

break;

case 5:

peek();

break;

case 6:

display();

break;

default:

cout<<"invalid choice";

}

}

return 0;

}

Q2

#include <iostream>

using namespace std;

#define size 10

int queue[size];

int f=0;

int r=0;

int isfull(){

if((r+1)%size==f){

return 1;

}

return 0;

}

int isempty(){

if(f==r){

return 1;

}

return 0;

}

void enqueue(int val){

if(isfull()){

cout<<"queue is full"<<endl;

}

else {

r=(r+1)%size;

queue[r]=val;

}

}

void dequeue(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

f=(f+1)%size;

cout<<queue[f]<<"dequeued"<<endl;

}

}

void display(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

int i = f+1;

while(true){

cout << queue[i] << " ";

if(i == r) break;

i = (i+1) % size;

}

}

}

void peek(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

cout<<"front most element is:"<<queue[f+1];

}

}

int main()

{

int value;

int choice;

while(true){

cout << "\n-- Queue Menu --\n";

cout << "1. Enqueue \n2.Dequeue \n3. isempty\n4. isfull\n5. Peek\n6. Display\n";

cout<<"enter your choice:"<<endl;

cin>>choice;

switch(choice){

case 1:

cout<<"enter value you want to enqueue:"<<endl;

cin>>value;

enqueue(value);

break;

case 2:

dequeue();

break;

case 3:

isempty();

break;

case 4:

isfull();

break;

case 5:

peek();

break;

case 6:

display();

break;

default:

cout<<"invalid choice";

}

}

return 0;

}

Q3

#include <iostream>

using namespace std;

#define size 6

int queue[size];

int f=-1;

int r=-1;

int arr1[size];

int isfull(){

if(r==size-1){

return 1;

}

return 0;

}

int isempty(){

if(f==r){

return 1;

}

return 0;

}

void enqueue(int val){

if(isfull()){

cout<<"queue is full"<<endl;

}

else {

r++;

queue[r]=val;

}

}

int dequeue(){

int a;

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

f++;

a=queue[f];

}

return a;

}

void display(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

for(int i=f+1;i<size;i++){

cout<<queue[i]<<endl;

}

}

}

void peek(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

cout<<"front most element is:"<<queue[f+1];

}

}

void interleave(){

int s=size/2;

int arr[s];

for(int i=0;i<size/2;i++){

arr[i]=dequeue();

}

for(int i=0;i<size/2;i++){

arr1[2\*i]=arr[i];

arr1[2\*i+1]=dequeue();

}

}

void show(){

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

cout<<arr1[i]<<endl;

}

}

int main()

{

enqueue(4);

enqueue(7);

enqueue(11);

enqueue(20);

enqueue(5);

enqueue(9);

interleave();

show();

return 0;

}

OR

#include <iostream>

using namespace std;

#define size 7

int a;

int queue[size];

int f=0;

int r=0;

int isfull(){

if((r+1)%size==f){

return 1;

}

return 0;

}

int isempty(){

if(f==r){

return 1;

}

return 0;

}

void enqueue(int val){

if(isfull()){

cout<<"queue is full"<<endl;

}

else {

r=(r+1)%size;

queue[r]=val;

}

}

int dequeue(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

f=(f+1)%size;

a=queue[f];

}

return a;

}

void display(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

int i = f+1;

while(true){

cout << queue[i] << " ";

if(i == r) break;

i = (i+1) % size;

}

}

}

void peek(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

cout<<"front most element is:"<<queue[f+1];

}

}

void interleave(){

int s=size/2;

int arr[s];

for(int i=0;i<6/2;i++){

arr[i]=dequeue();

}

for(int i=0;i<6/2;i++){

enqueue(arr[i]);

enqueue(dequeue());

}

}

int main()

{

enqueue(4);

enqueue(7);

enqueue(11);

enqueue(20);

enqueue(5);

enqueue(9);

interleave();

display();

return 0;

}

Q4

#include <iostream>

#include<string>

using namespace std;

#define size 20

int a;

char queue[size];

int f=-1;

int r=-1;

int isfull(){

if(r==size-1){

return 1;

}

return 0;

}

int isempty(){

if(f==r){

return 1;

}

return 0;

}

void enqueue(int val){

if(isfull()){

cout<<"queue is full"<<endl;

}

else {

r++;

queue[r]=val;

}

}

void dequeue(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

f++;

}

}

void display(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

for(int i=f+1;i<size;i++){

cout<<queue[i]<<endl;

}

}

}

int peek(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

a=queue[f+1];

}

return a;

}

int main()

{

string s;

cout<<"enter the string";

cin>>s;

int freq[256]={0};

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

char ch=s[i];

freq[ch]++;

enqueue(ch);

while(!isempty()&& freq[peek()]){

dequeue();

}

if(!isempty())

{

cout<<peek()<<endl;

}

else

{

cout<<"-1";

}

}

return 0;

}

Q5

(a)

#include <iostream>

using namespace std;

#define size 10

struct queue{

int f=-1;

int r=-1;

int arr[size];

int isfull(){

if(r==size-1){

return 1;

}

return 0;

}

int isempty(){

if(f==-1||f>r){

return 1;

}

return 0;

}

void enqueue(int val){

if(isfull()){

cout<<"queue is full"<<endl;

}

else if(isempty()){

r=0;

f=0;

arr[r]=val;

}

else

r++;

arr[r]=val;

}

void dequeue(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

f++;

}

}

void display(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

for(int i=f+1;i<=r;i++){

cout<<arr[i]<<endl;

}

}

}

int peek(){

int a;

if(isempty()){

cout<<"queue is empty"<<endl;

return -1;

}

return arr[f];

}

};

queue q1;

queue q2;

void push(int x){

while(!q1.isempty()){

q2.enqueue(q1.peek());

q1.dequeue();

}

q1.enqueue(x);

while(!q2.isempty()){

q1.enqueue(q2.peek());

q2.dequeue();

}

}

int pop(){

int b;

b=q1.peek();

q1.dequeue();

return b;

}

int top(){

return q1.peek();

}

int empty(){

if(q1.isempty()){

return 1;

}

else

return 0;

}

int main()

{

push(10);

push(20);

push(30);

cout << "Top element: " << top() << endl;

cout << "Pop: " << pop() << endl;

cout << "Top after pop: " << top() << endl;

return 0;

}

(b)

#include <iostream>

using namespace std;

#define MAXLEN 100

struct queue{

int f=-1;

int r=-1;

int arr[MAXLEN];

int isfull(){

if(r==MAXLEN-1){

return 1;

}

return 0;

}

int isempty(){

if(f==-1||f>r){

return 1;

}

return 0;

}

void enqueue(int val){

if(isfull()){

cout<<"queue is full"<<endl;

}

else if(isempty()){

r=0;

f=0;

arr[r]=val;

}

else

r++;

arr[r]=val;

}

void dequeue(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

f++;

}

}

void display(){

if(isempty()){

cout<<"queue is empty"<<endl;

}

else{

for(int i=f;i<=r;i++){

cout<<arr[i]<<endl;

}

}

}

int peek(){

int a;

if(isempty()){

cout<<"queue is empty"<<endl;

return -1;

}

return arr[f];

}

int size(){

return r-f+1;

}

};

queue q1;

void push(int x){

q1.enqueue(x);

for(int i=0;i<q1.size()-1;i++){

q1.enqueue(q1.peek());

q1.dequeue();

}

}

int pop(){

int b;

b=q1.peek();

q1.dequeue();

return b;

}

int top(){

return q1.peek();

}

int empty(){

if(q1.isempty()){

return 1;

}

else

return 0;

}

int main()

{

push(10);

push(20);

push(30);

cout << "Top element: " << top() << endl;

cout << "Pop: " << pop() << endl;

cout << "Top after pop: " << top() << endl;

return 0;

}