NAME:KHUSHI SHAH 53 SYIT

EXPT 4

CODE:-

#include <stdio.h>

#define MAX 10

int deque[MAX];

int left = -1, right = -1;

void input\_deque(void);

void output\_deque(void);

void insert\_left(void);#include <stdio.h>

#define MAX 10

int deque[MAX];

int left = -1, right = -1;

void input\_deque(void);

void output\_deque(void);

void insert\_left(void);

void insert\_right(void);

void delete\_left(void);

void delete\_right(void);

void display(void);

int main() {

int option;

printf("\n \*\*\*\*\*MAIN MENU\*\*\*\*\*");

printf("\n 1.Input restricted deque");

printf("\n 2.Output restricted deque");

printf("\n Enter your option : ");

scanf("%d", &option);

switch (option) {

case 1:

input\_deque();

break;

case 2:

output\_deque();

break;

}

return 0;

}

void input\_deque() {

int option;

do {

printf("\n INPUT RESTRICTED DEQUE");

printf("\n 1.Insert at right");

printf("\n 2.Delete from left");

printf("\n 3.Delete from right");

printf("\n 4.Display");

printf("\n 5.Quit");

printf("\n Enter your option : ");

scanf("%d", &option);

switch (option) {

case 1:

insert\_right();

break;

case 2:

delete\_left();

break;

case 3:

delete\_right();

break;

case 4:

display();

break;

}

} while (option != 5);

}

void output\_deque() {

int option;

do {

printf("OUTPUT RESTRICTED DEQUE");

printf("\n 1.Insert at right");

printf("\n 2.Insert at left");

printf("\n 3.Delete from left");

printf("\n 4.Display");

printf("\n 5.Quit");

printf("\n Enter your option : ");

scanf("%d", &option);

switch (option) {

case 1:

insert\_right();

break;

case 2:

insert\_left();

break;

case 3:

delete\_left();

break;

case 4:

display();

break;

}

} while (option != 5);

}

void insert\_right() {

int val;

printf("\n Enter the value to be added: ");

scanf("%d", &val);

if ((left == 0 && right == MAX - 1) || (left == right + 1)) {

printf("\n OVERFLOW");

return;

}

if (left == -1) {

/\* If queue is initially empty \*/

left = 0;

right = 0;

} else {

if (right == MAX - 1) /\* right is at the last position of queue \*/

right = 0;

else

right = right + 1;

}

deque[right] = val;

}

void insert\_left() {

int val;

printf("\n Enter the value to be added: ");

scanf("%d", &val);

if ((left == 0 && right == MAX - 1) || (left == right + 1)) {

printf("\n Overflow");

return;

}

if (left == -1) {

/\* If queue is initially empty \*/

left = 0;

right = 0;

} else {

if (left == 0)

left = MAX - 1;

else

left = left - 1;

}

deque[left] = val;

}

void delete\_left() {

if (left == -1) {

printf("\n UNDERFLOW");

return;

}

printf("\n The deleted element is: %d", deque[left]);

if (left == right) /\* Queue has only one element \*/

{

left = -1;

right = -1;

} else {

if (left == MAX - 1)

left = 0;

else

left = left + 1;

}

}

void delete\_right() {

if (left == -1) {

printf("\n UNDERFLOW");

return;

}

printf("\n The element deleted is: %d", deque[right]);

if (left == right) /\* Queue has only one element \*/

{

left = -1;

right = -1;

} else {

if (right == 0)

right = MAX - 1;

else

right = right - 1;

}

}

void display() {

int front = left, rear = right;

if (front == -1) {

printf("\n QUEUE IS EMPTY");

return;

}

printf("\n The elements of the queue are: ");

if (front <= rear) {

while (front <= rear) {

printf("%d ", deque[front]);

front++;

}

} else {

while (front <= MAX - 1) {

printf("%d ", deque[front]);

front++;

}

front = 0;

while (front <= rear) {

printf("%d ", deque[front]);

front++;

}

}

printf("\n");

}

void insert\_right(void);

void delete\_left(void);

void delete\_right(void);

void display(void);

int main() {

int option;

printf("\n \*\*\*\*\*MAIN MENU\*\*\*\*\*");

printf("\n 1.Input restricted deque");

printf("\n 2.Output restricted deque");

printf("\n Enter your option : ");

scanf("%d", &option);

switch (option) {

case 1:

input\_deque();

break;

case 2:

output\_deque();

break;

}

return 0;

}

void input\_deque() {

int option;

do {

printf("\n INPUT RESTRICTED DEQUE");

printf("\n 1.Insert at right");

printf("\n 2.Delete from left");

printf("\n 3.Delete from right");

printf("\n 4.Display");

printf("\n 5.Quit");

printf("\n Enter your option : ");

scanf("%d", &option);

switch (option) {

case 1:

insert\_right();

break;

case 2:

delete\_left();

break;

case 3:

delete\_right();

break;

case 4:

display();

break;

}

} while (option != 5);

}

void output\_deque() {

int option;

do {

printf("OUTPUT RESTRICTED DEQUE");

printf("\n 1.Insert at right");

printf("\n 2.Insert at left");

printf("\n 3.Delete from left");

printf("\n 4.Display");

printf("\n 5.Quit");

printf("\n Enter your option : ");

scanf("%d", &option);

switch (option) {

case 1:

insert\_right();

break;

case 2:

insert\_left();

break;

case 3:

delete\_left();

break;

case 4:

display();

break;

}

} while (option != 5);

}

void insert\_right() {

int val;

printf("\n Enter the value to be added: ");

scanf("%d", &val);

if ((left == 0 && right == MAX - 1) || (left == right + 1)) {

printf("\n OVERFLOW");

return;

}

if (left == -1) {

/\* If queue is initially empty \*/

left = 0;

right = 0;

} else {

if (right == MAX - 1) /\* right is at the last position of queue \*/

right = 0;

else

right = right + 1;

}

deque[right] = val;

}

void insert\_left() {

int val;

printf("\n Enter the value to be added: ");

scanf("%d", &val);

if ((left == 0 && right == MAX - 1) || (left == right + 1)) {

printf("\n Overflow");

return;

}

if (left == -1) {

/\* If queue is initially empty \*/

left = 0;

right = 0;

} else {

if (left == 0)

left = MAX - 1;

else

left = left - 1;

}

deque[left] = val;

}

void delete\_left() {

if (left == -1) {

printf("\n UNDERFLOW");

return;

}

printf("\n The deleted element is: %d", deque[left]);

if (left == right) /\* Queue has only one element \*/

{

left = -1;

right = -1;

} else {

if (left == MAX - 1)

left = 0;

else

left = left + 1;

}

}

void delete\_right() {

if (left == -1) {

printf("\n UNDERFLOW");

return;

}

printf("\n The element deleted is: %d", deque[right]);

if (left == right) /\* Queue has only one element \*/

{

left = -1;

right = -1;

} else {

if (right == 0)

right = MAX - 1;

else

right = right - 1;

}

}

void display() {

int front = left, rear = right;

if (front == -1) {

printf("\n QUEUE IS EMPTY");

return;

}

printf("\n The elements of the queue are: ");

if (front <= rear) {

while (front <= rear) {

printf("%d ", deque[front]);

front++;

}

} else {

while (front <= MAX - 1) {

printf("%d ", deque[front]);

front++;

}

front = 0;

while (front <= rear) {

printf("%d ", deque[front]);

front++;

}

}

printf("\n");

}



