```
/* Write a C program to implement (perform) double ended queue (deque). */

#include <stdio.h>
#include <conio.h>

void insfront();
void insrear();
void delfront();
void delrear();
void display();

int deque[5], front = -1, rear = -1;

void main()
{
	int choice = 0;
	clrscr();
	while(choice != 6)
	{
		clrscr();
		printf("\n Main Menu (Operations on Deque)");
		printf("\n 1. INSERT AT FRONT END");
		printf("\n 2. INSERT AT REAR END");
		printf("\n 3. DELETE AT FRONT END");
		printf("\n 4. DELETE AT REAR END");
		printf("\n 5. DISPLAY");
		printf("\n 6. EXIT");
		printf("\n Enter your choice: ");
		scanf("%d", &choice);
		switch(choice)
		{
			case 1:
				insfront();
				break;
			case 2:
				insrear();
				break;
			case 3:
				delfront();
				break;
			case 4:
				delrear();
				break;
			case 5:
				display();
				break;
			case 6:
				break;
			default:
				printf("\n Invalid Choice");
		}
		printf("\n Press any key to continue...");
		getch();
	}
}

void insfront()
{
	int value = 0;
	if(front == 0)
	{
		printf("\n Queue is Full (Queue Overflow)");
	}
	else
	{
		printf("\n Enter a value to insert at front end: ");
```

```
                scanf("%d", &value);
                if(front == -1)
                {
                        front = 0;
                        rear = 0;
                }
                else
                {
                        front = front - 1;
                }
                deque[front] = value;
        }
}

void insrear()
{
        int value = 0;
        if(rear == 4)
        {
                printf("\n Queue is Full (Queue Overflow)");
        }
        else
        {
                printf("\n Enter a value to insert at rear end: ");
                scanf("%d", &value);
                rear = rear + 1;
                deque[rear] = value;
                if(front == -1)
                {
                        front = 0;
                }
        }
}

void delfront()
{
        int value = 0;
        if(front == -1)
        {
                printf("\n Queue is Empty (Queue Unserflow)");
        }
        else
        {
                value = deque[front];
                if(front == rear)
                {
                        front = -1;
                        rear = -1;
                }
                else
                {
                        front = front + 1;
                }
                printf("\n Value deleted: %d", value);
        }
}

void delrear()
{
        int value = 0;
        if(front == -1)
        {
                printf("\n Queue is Empty (Queue Unserflow)");
        }
        else
        {
```

```
                value = deque[rear];
                if(front == rear)
                {
                        front = -1;
                        rear = -1;
                }
                else
                {
                        rear = rear - 1;
                }
                printf("\n Value deleted: %d", value);
        }
}

void display()
{
        int i = 0;
        if(front == -1)
        {
                printf("\n Queue is Empty (Queue Unserflow)");
        }
        else
        {
                printf("\n Values in the queue are: ");
                for(i = front; i <= rear; i++)
                {
                        printf("\n %d", deque[i]);
                }
        }
}
```