```
//============================================================================
// Name        : 21465_Pract13.cpp
// Author      : Chaitanya Paraskar
// Roll No.    : 21465
// Aim         : A double-ended queue (deque) is a linear list in which
additions and deletions may be
//               made at either end.Obtain a data representation mapping a deque
into a one dimensional array.
//               Write C++ program to simulate deque with functions to add and
delete elements from either
//               end of the deque.
//============================================================================

#include <iostream>
using namespace std;

class deque
{
    int size;
    int *arr;
    int front;
    int back;

public:
    deque(int _size) : front(-1), back(-1), size(_size)
    {
        arr = new int[_size];
    }

    void push_back(int _data)
    {
        if ((front == 0 && back == size) || (front == back + 1))
        {
            cout << "Deque OverFlow" << endl;
            return;
        }
        else if (front == -1)
            front = back = 0;
        else if (front != 0 && back == size)
            back = 0;
        else
            back++;

        arr[back] = _data;
    }

    void push_front(int _data)
    {
        if ((front == 0 && back == size) || (front == back + 1))
        {
            cout << "Deque OverFlow" << endl;
            return;
        }
        else if (front == -1)
```

```
        front = back = 0;
    else if (front == 0 && back != size)
        front = size;
    else
        front--;
    arr[front] = _data;
}

void pop_back()
{
    if (front == -1)
        cout << "Deque is Empty" << endl;
    else if (front == back)
        front = back = -1;
    else if (back == 0)
        back = size;
    else
        back--;
}

void pop_front()
{
    if (front == -1)
        cout << "Deque is Empty" << endl;
    else if (front == back)
        front = back = -1;
    else if (front == size)
        front = 0;
    else
        front++;
}

int getFront()
{
    if (front == -1)
    {
        cout << "Deque if Empty" << endl;
        return -1;
    }
    else
        return arr[front];
}

void print()
{
    if (front == -1)
    {
        cout << "Deque is empty" << endl;
        return;
    }
    else
    {
        int start = front;
        int end = back;
```

```
            while (start != back)
            {
                cout << arr[start] << " ";
                if (start == size)
                    start = 0;
                else
                    start++;
            }
            cout << arr[end] << endl;
        }
    }

    ~deque()
    {
        cout << "Deque Deleted" << endl;
        delete[] arr;
    }
};

int main()
{
    int dequeSize = 10;

    cout << "Size of Queue = 10";

    deque myDeque(dequeSize - 1);

    int choice;
    do
    {
        cout << "\nDeque Menu:\n";
        cout << "1. Display Deque from Front to Back\n";
        cout << "2. Push Front\n";
        cout << "3. Push Back\n";
        cout << "4. Pop Front\n";
        cout << "5. Pop Back\n";
        cout << "6. Exit\n";

        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice)
        {
        case 1:
            myDeque.print();
            break;

        case 3:
            int dataPushBack;
            cout << "Enter data to push back: ";
            cin >> dataPushBack;
            myDeque.push_back(dataPushBack);
            myDeque.print();
            break;
```

```
        case 2:
            int dataPushFront;
            cout << "Enter data to push front: ";
            cin >> dataPushFront;
            myDeque.push_front(dataPushFront);
            myDeque.print();
            break;

        case 5:
            myDeque.pop_back();
            myDeque.print();
            break;

        case 4:
            myDeque.pop_front();
            myDeque.print();
            break;

        case 6:
            cout << "Exiting Deque Menu\n";
            break;

        default:
            cout << "Invalid choice. Please enter a valid option.\n";
        }

    } while (choice != 6);

    return 0;
}

/*

Output:

$ g++ Praact13.cpp -o out && ./out
Size of Queue = 10
Deque Menu:
1. Display Deque from Front to Back
2. Push Front
3. Push Back
4. Pop Front
5. Pop Back
6. Exit
Enter your choice: 1
Deque is empty

Deque Menu:
1. Display Deque from Front to Back
2. Push Front
3. Push Back
4. Pop Front
5. Pop Back
```

```
6. Exit
Enter your choice: 2
Enter data to push front: 123
123

Deque Menu:
1. Display Deque from Front to Back
2. Push Front
3. Push Back
4. Pop Front
5. Pop Back
6. Exit
Enter your choice: 2
Enter data to push front: 234
234 123

Deque Menu:
1. Display Deque from Front to Back
2. Push Front
3. Push Back
4. Pop Front
5. Pop Back
6. Exit
Enter your choice: 2
Enter data to push front: 3
3 234 123

Deque Menu:
1. Display Deque from Front to Back
2. Push Front
3. Push Back
4. Pop Front
5. Pop Back
6. Exit
Enter your choice: 3
Enter data to push back: 456
3 234 123 456

Deque Menu:
1. Display Deque from Front to Back
2. Push Front
3. Push Back
4. Pop Front
5. Pop Back
6. Exit
Enter your choice: 4
234 123 456

Deque Menu:
1. Display Deque from Front to Back
2. Push Front
3. Push Back
4. Pop Front
5. Pop Back
```

```
6. Exit
Enter your choice: 5
234 123

Deque Menu:
1. Display Deque from Front to Back
2. Push Front
3. Push Back
4. Pop Front
5. Pop Back
6. Exit
Enter your choice: 4
123

Deque Menu:
1. Display Deque from Front to Back
2. Push Front
3. Push Back
4. Pop Front
5. Pop Back
6. Exit
Enter your choice: 5
Deque is empty

Deque Menu:
1. Display Deque from Front to Back
2. Push Front
3. Push Back
4. Pop Front
5. Pop Back
6. Exit
Enter your choice: 6
Exiting Deque Menu
Deque Deleted

*/
```