**Array Implementation of Queue**

*Source Code:*

#include <stdio.h>

#include <conio.h>

#define MAX 10 // Changing this value will change length of array

int queue[MAX];

int front = -1, rear = -1;

void insert(void);

int delete\_element(void);

int peek(void);

void display(void);

int main()

{

    int option, val;

    do

    {

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

        printf("\n 1. Insert an element");

        printf("\n 2. Delete an element");

        printf("\n 3. Peek");

        printf("\n 4. Display the queue");

        printf("\n 5. EXIT");

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

        scanf("%d", &option);

        switch(option)

        {

            case 1:

                insert();

                break;

            case 2:

                val = delete\_element();

                if (val != -1)

                    printf("\n The number deleted is : %d", val);

                break;

            case 3:

                val = peek();

                if (val != -1)

                    printf("\n The first value in queue is : %d", val);

                break;

            case 4:

                display();

                break;

        }

    } while(option != 5);

    getch();

    return 0;

}

void insert()

{

    int num;

    printf("\n Enter the number to be inserted in the queue : ");

    scanf("%d", &num);

    if(rear == MAX-1)

        printf("\n OVERFLOW");

    else if(front == -1 && rear == -1)

        front = rear = 0;

    else

        rear++;

    queue[rear] = num;

}

int delete\_element()

{

    int val;

    if(front == -1 || front > rear)

    {

        printf("\n UNDERFLOW");

        return -1;

    }

    else

    {

        val = queue[front];

        front++;

        if(front > rear)

            front = rear = -1;

        return val;

    }

}

int peek()

{

    if(front == -1 || front > rear)

    {

        printf("\n QUEUE IS EMPTY");

        return -1;

    }

    else

    {

        return queue[front];

    }

}

void display()

{

    int i;

    printf("\n");

    if(front == -1 || front > rear)

        printf("\n QUEUE IS EMPTY");

    else

    {

        for(i = front; i <= rear; i++)

            printf("\t %d", queue[i]);

    }

}

*Output:*

\*\*\*\*\* MAIN MENU \*\*\*\*\*

1. Insert an element

2. Delete an element

3. Peek

4. Display the queue

5. EXIT

Enter your option : 1

Enter the number to be inserted in the queue : 12

\*\*\*\*\* MAIN MENU \*\*\*\*\*

1. Insert an element

2. Delete an element

3. Peek

4. Display the queue

5. EXIT

Enter your option : 1

Enter the number to be inserted in the queue : 13

\*\*\*\*\* MAIN MENU \*\*\*\*\*

1. Insert an element

2. Delete an element

3. Peek

4. Display the queue

5. EXIT

Enter your option : 2

The number deleted is : 0

\*\*\*\*\* MAIN MENU \*\*\*\*\*

1. Insert an element

2. Delete an element

3. Peek

4. Display the queue

5. EXIT

Enter your option : 4

12

\*\*\*\*\* MAIN MENU \*\*\*\*\*

1. Insert an element

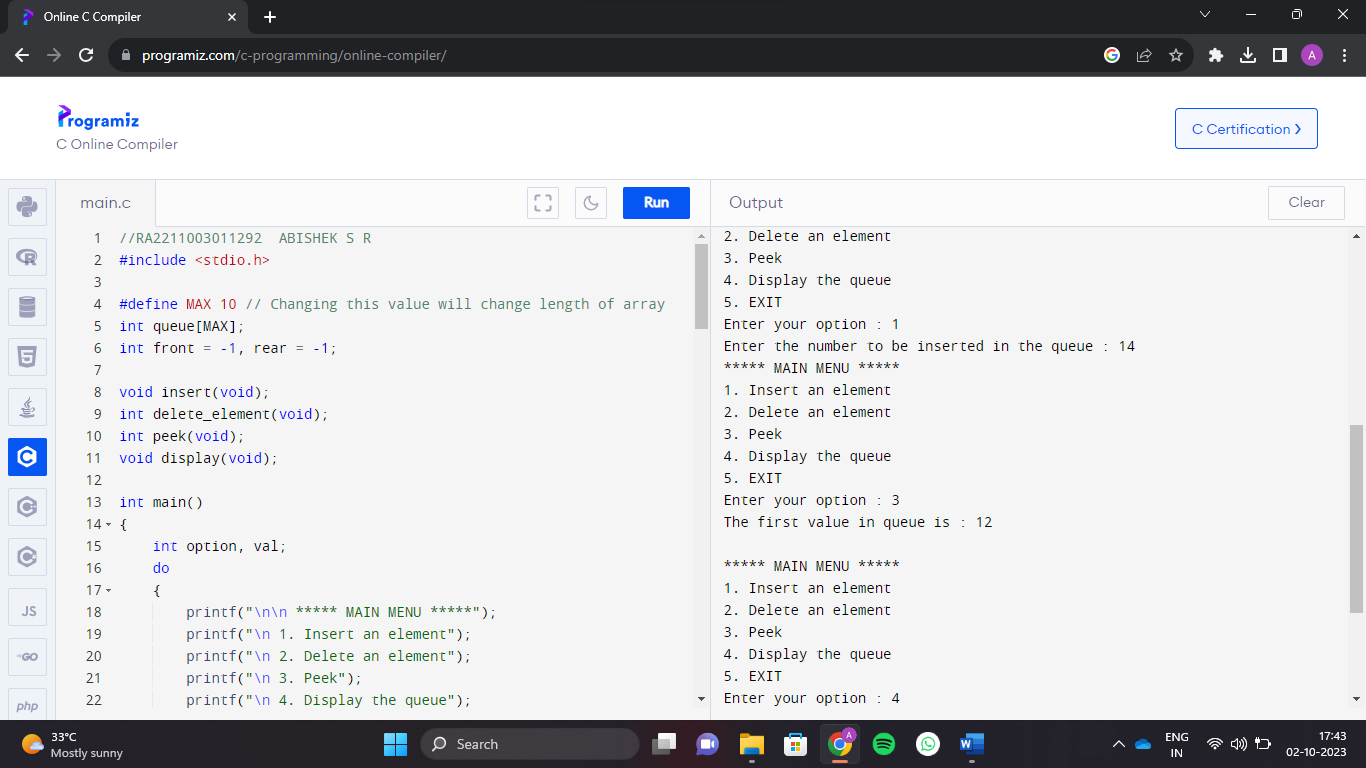
2. Delete an element

3. Peek

4. Display the queue

5. EXIT

Enter your option : 5

****

**ABISHEK S R  
RA2211003011292**