

DSA

INTERNSHALA

Implementation of Stack and Queue

Stack

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#define MAX 5

typedef struct stack_node {
    int arr[MAX];
    int top;
} node;

void push(int item, node *ptr) {
    if(ptr->top == MAX-1) {
        printf("\n\n\t ***** Overflow *****");
        printf("\n\n\t Element can't be inserted into Stack");
    } else {
        ptr-> top++;
        ptr->arr[ptr->top] = item;
        printf("\n\n\t Element is pushed into Stack");
    }
    getch();
}

void pop(node *ptr) {
    int item;
    if(ptr->top < 0) {
        printf("\n\n\t ***** Underflow *****");
    } else {
        item = ptr-> arr[ptr->top];
        ptr->top--;
        printf("\n\n\t Element [%d] is popped from Stack",item);
    }
    getch();
    return;
}

int main() {
    int i , item;
    char ch;
    node stack;
    stack.top = -1;
```

```

while(1) {
    system("cls"); // clear screen
    printf("\n\n\t ***** Main Menu *****\n");
    printf("\n\t Array representation of stack");
    printf("\n\t =====");
    printf("\n\n\t 1) Push operation on Stack");
    printf("\n\n\t 2) Pop operation on Stack");
    printf("\n\n\t 3) Exit");
    printf("\n\n\t Enter your Choice :");
    fflush(stdin);
    scanf("%c",&ch);
    switch(ch) {
        case '1' :
            printf("\n\n\t Enter the no. you want to add :");
            scanf("%d",&item);
            push(item,&stack);
            break;
        case '2' :
            pop(&stack);
            break;
        case '3' :
            exit(0);
    }
}
return 0;
}

```

Queue

```

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define MX 5

typedef struct queue_type {
    int arr[MX];
    int fornt;
    int rear;
} node;

void insert(node *q, int item) {
    if((q->front==0 && q->rear==MX-1) || q->front==q->rear+1) {
        printf("Queue Overflow error.....");
        getch();
        return;
    }
    if(q->front == -1) {
        q->front = q->rear = 0;
    } else if(q->rear == MX-1) {
        q->rear = 0
    }
}

```

```

    } else {
        q->rear=q->rear+1;
    }
    q->arr[q->rear] = item;
}

```

```

int delete(node *q) {
    int val;
    if(q->front == -1) {
        printf("Queue underflow error.....");
        getch();
        return;
    }
    val = q->arr[q->front];
    if(q->front == q->rear) {
        q->front = q->rear = -1;
    } else if(q->front == MX-1) {
        q->front = 0;
    } else {
        q->front++;
    }
    return val;
}

```

```

int main() {
    char ch;
    int val, item;
    node queue;
    queue.front = -1;
    queue.rear = -1;

```

```

    while(1) {
        system("cls");
        printf("\n\n\t **** Main Block **** \n");
        printf("\n\t Array representation of Queue");
        printf("\n\t=====");
        printf("\n\n\t 1) Inserting element into Queue");
        printf("\n\n\t 2) Deleting element form Queue");
        printf("\n\n\t 3) Exit from program");
        printf("\n\n\t Enter your Choice :➔ ");
        fflush(stdin);
        scanf("%c",&ch);
        switch(ch) {
            case '1' :
                printf("\n\n\t Enter the Number to Push :=> ");
                scanf("%d",&item);
                insert(&queue,item);
                break;
            case '2' :
                val = delete(&queue);
                printf("The value of deleted element is %d",val);
                break;

```

```
    case '3' :  
        exit(0);  
    }  
}  
return 0;  
}
```