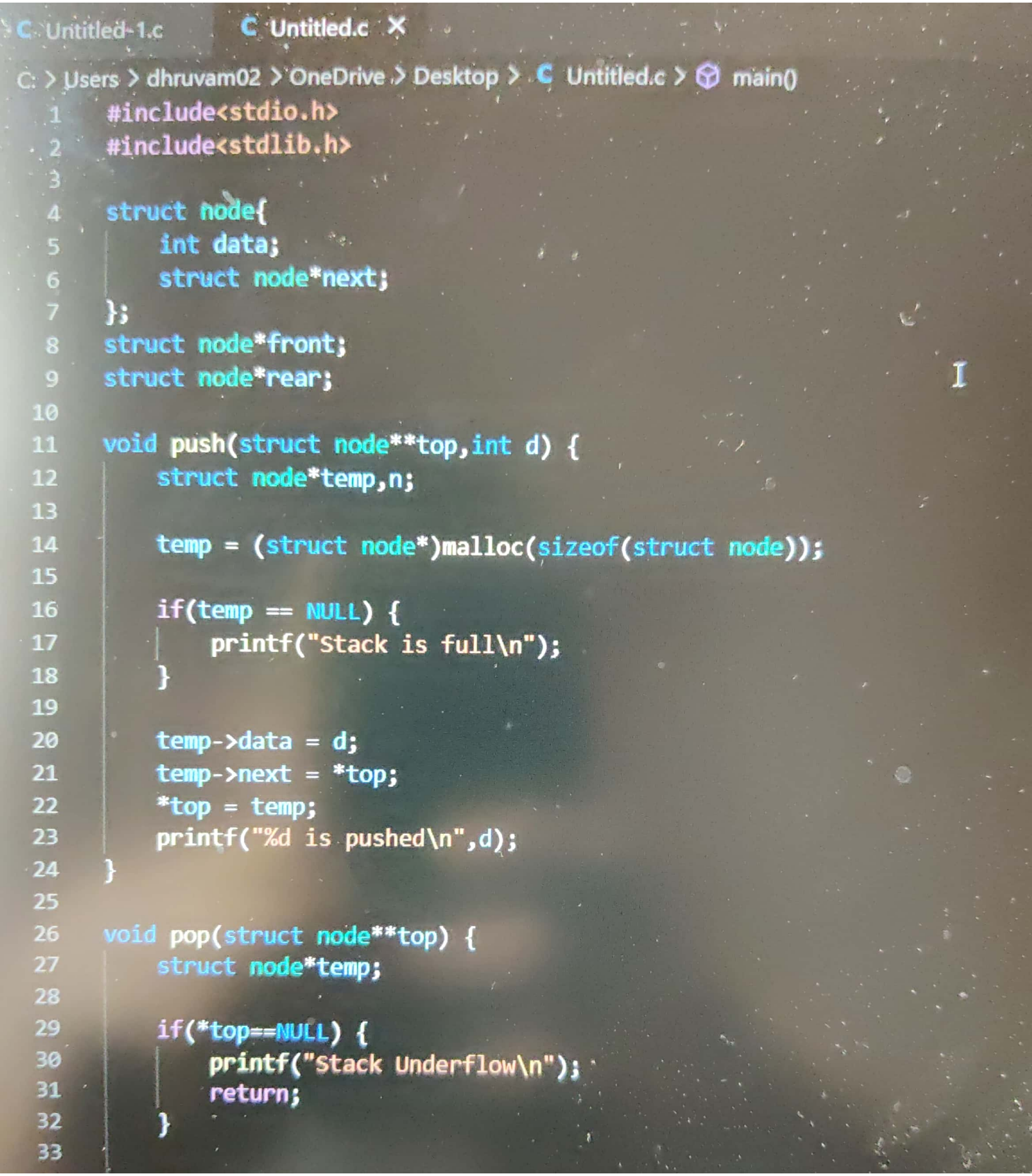
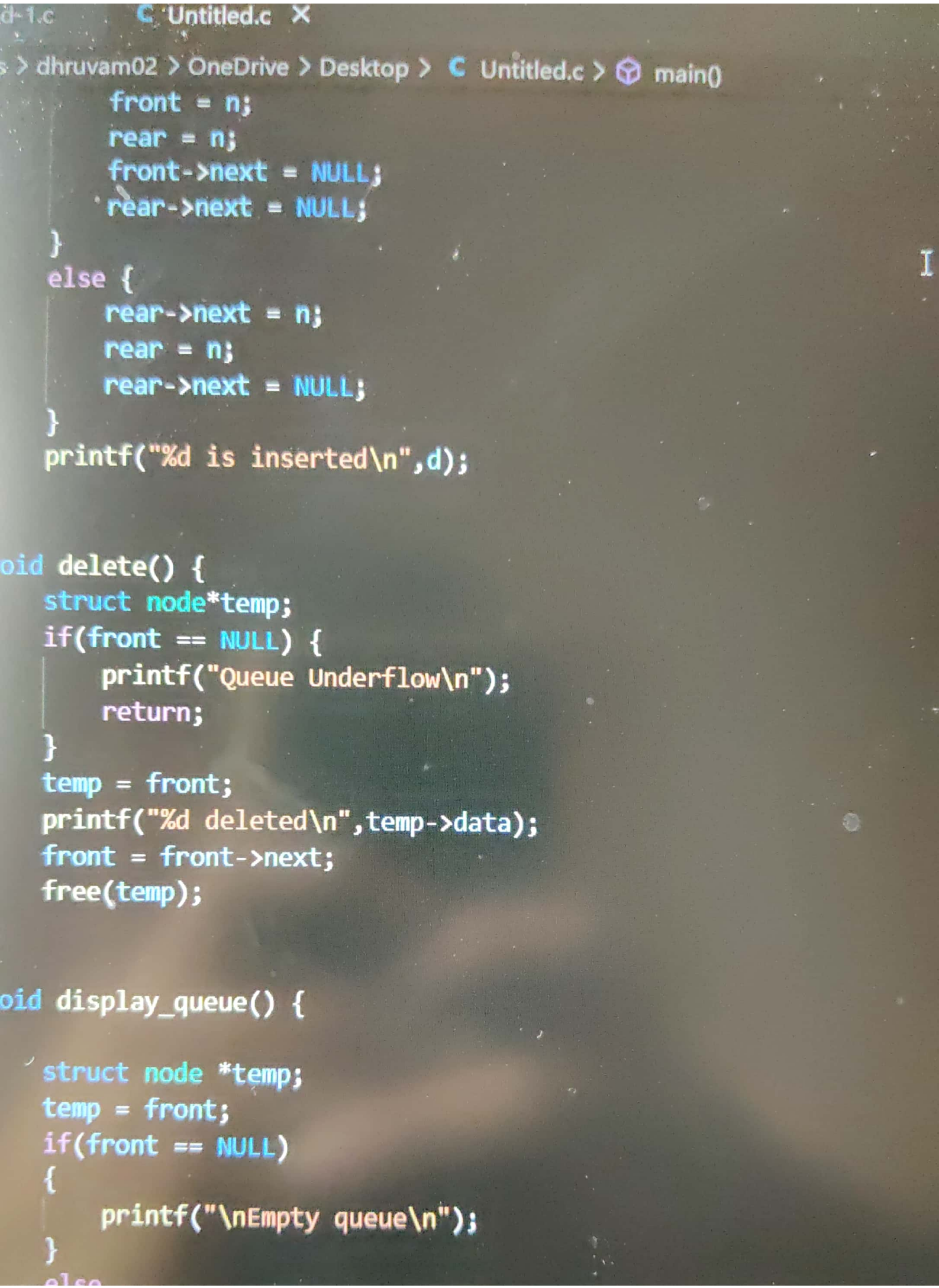


1.c C Untitled.c ×

> dhruvam02 > OneDrive > Desktop > C Untitled.c > main()

```
        printf("Stack Underflow\n");
        return;
    }

    temp = *top;
    printf("%d poped\n",temp->data);
    *top = (*top)->next;

    free(temp);


oid display(struct node* top) {
    if(top == NULL){
        printf("No Elements Present in Stack\n");
        return;
    }

    while(top!=NULL) {
        printf("%d ",top->data);
        top = top->next;
    }
    printf("\n");
}

void insert(int d) {
    struct node*n;
    n = (struct node*)malloc(sizeof(struct node));
    if(n == NULL){
        printf("Queue Overflow\n");
        return;
    }
    n->data = d;
    if(front==NULL) {
        front = n;
```



```
d-1.c        C Untitled.c ×
s > dhruvam02 > OneDrive > Desktop > C Untitled.c > main()
        front = n;
        rear = n;
        front->next = NULL;
        rear->next = NULL;
    }
    else {
        rear->next = n;
        rear = n;
        rear->next = NULL;
    }
    printf("%d is inserted\n",d);


oid delete() {
    struct node*temp;
    if(front == NULL) {
        printf("Queue Underflow\n");
        return;
    }
    temp = front;
    printf("%d deleted\n",temp->data);
    front = front->next;
    free(temp);


oid display_queue() {

    struct node *temp;
    temp = front;
    if(front == NULL)
    {
        printf("\nEmpty queue\n");
    }
    else
```

```
    else
    {   printf("\nQueue Elements: \n");
        while(temp != NULL)
        {
            printf("%d ",temp -> data);
            temp = temp -> next;
        }
        printf("\n");
    }
}

int main() {
    struct node*stack = NULL;
    printf("STACK OPERATIONS\n");
    printf("1.Push\t2.Pop\t3.Display\t4.Exit\n");
    int choice,item;
    printf("Enter your choice: ");
    scanf("%d",&choice);
    while(choice!=4) {
        switch(choice) {
            case 1: printf("Enter data to be pushed: ");
                    scanf("%d",&item);
                    push(&stack,item);
                    break;

            case 2: pop(&stack);
                    break;

            case 3: display(stack);
                    break;
        }
        printf("1.Push\t2.Pop\t3.Display\t4.Exit\n");
        printf("Enter your choice: ");
```

```
            break;
        }
        printf("1.Push\t2.Pop\t3.Display\t4.Exit\n");
        printf("Enter your choice: ");
        scanf("%d",&choice);
    }
    printf("End of Stack Operations\n\n");

    printf("QUEUE OPERATIONS\n");
    printf("1.Insert\t2.Delete\t3.Display\t4.Exit\n");
    printf("Enter your choice: ");
    scanf("%d",&choice);
    while(choice!=4) {
        switch(choice) {
            case 1: printf("Enter data to be inserted: ");
                    scanf("%d",&item);
                    insert(item);
                    break;

            case 2: delete();
                    break;

            case 3: display_queue();
                    break;
        }
        printf("1.Insert\t2.Delete\t3.Display\t4.Exit\n");
        printf("Enter your choice: ");
        scanf("%d",&choice);
    }
    printf("End Of Queue Operations\n");
    return 0;
}
```