

```
import java.util.Scanner;

class ToDoItem {
    private String description;
    private int priority;

    public ToDoItem(String description, int priority) {
        this.description = description;
        this.priority = priority;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public int getPriority() {
        return priority;
    }

    public void setPriority(int priority) {
        this.priority = priority;
    }
}

class ToDoList {
    private static final int MAX_SIZE = 10;
    private ToDoItem[] items;
    private int size;

    public ToDoList() {
        items = new ToDoItem[MAX_SIZE];
        size = 0;
    }

    public void addItem(String description, int priority) {
        ToDoItem newItem = new ToDoItem(description, priority);
        if (size == 0) {
            items[size] = newItem;
        } else {
            int i;
            for (i = size - 1; i >= 0; i--) {
```

```

        if (items[i].getPriority() < priority) {
            items[i + 1] = items[i];
        } else {
            break;
        }
    }
    items[i + 1] = newItem;
}
size++;
}

public void removeHighestPriorityItem() {
    if (size > 0) {
        for (int i = 0; i < size - 1; i++) {
            items[i] = items[i + 1];
        }
        size--;
    } else {
        System.out.println("List is empty. No items to remove.");
    }
}

public void updateItemDescription(int priority, String newDescription) {
    int index = findItemIndexByPriority(priority);
    if (index != -1) {
        items[index].setDescription(newDescription);
    } else {
        System.out.println("Item with priority " + priority + " not found.");
    }
}

private int findItemIndexByPriority(int priority) {
    for (int i = 0; i < size; i++) {
        if (items[i].getPriority() == priority) {
            return i;
        }
    }
    return -1; // Not found
}

public void displayItems() {
    if (size == 0) {
        System.out.println("List is empty.");
    }
}

```

```

    } else {
        for (int i = 0; i < size; i++) {
            System.out.println(" Priority: " + items[i].getPriority() + ",
Description: " + items[i].getDescription());
        }
    }
}
}
}

```

```

class Sample { //using priority queue
    public static void main(String[] args) {
        ToDoList todoList = new ToDoList();

        while (true) {
            System.out.println("Choose an option:");
            System.out.println("1. Add a task");
            System.out.println("2. Remove the highest priority task");
            System.out.println("3. Update a task's description");
            System.out.println("4. Display tasks");
            System.out.println("5. Exit");
            System.out.println();

            Scanner sc = new Scanner(System.in);
            int choice = sc.nextInt();
            sc.nextLine(); // Consume the newline

            switch (choice) {
                case 1:
                    System.out.print("Enter task description: ");
                    String description = sc.nextLine();
                    System.out.print("Enter task priority: ");
                    int priority = sc.nextInt();
                    sc.nextLine(); // Consume the newline

                    todoList.addItem(description, priority);
                    System.out.println("Task added.");
                    break;

                case 2:
                    System.out.println("\nRemoving the highest priority item:");
                    todoList.removeHighestPriorityItem();
                    break;

                case 3:
                    System.out.println("\nUpdating an item's description:");

```

```
        System.out.println("Enter the priority of the task you want to
update: ");
        int updatePriority = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter the new task description: ");
        String newDescription = sc.nextLine();
        todoList.updateItemDescription(updatePriority, newDescription);
        todoList.displayItems();

        break;

        case 4:
        todoList.displayItems();
        break;

        case 5:
            sc.close();
            System.out.println("Exiting the program.");
            System.exit(0);

        default:
            System.out.println("Invalid choice. Please enter a valid
option.");
    }
}
}
```

```
PS E:\PG - DAC\DSA\Mini Project> java Sample
```

```
Choose an option:
```

1. Add a task
2. Remove the highest priority task
3. Update a task's description
4. Display tasks
5. Exit

```
1
```

```
Enter task description: Buy Fire Crackers
```

```
Enter task priority: 4
```

```
Task added.
```

```
Choose an option:
```

1. Add a task
2. Remove the highest priority task
3. Update a task's description
4. Display tasks
5. Exit

```
1
```

```
Enter task description: Buy Milk
```

```
Enter task priority: 3
```

```
Task added.
```

```
Choose an option:
```

1. Add a task
2. Remove the highest priority task
3. Update a task's description
4. Display tasks
5. Exit

```
1
```

```
Enter task description: Attend Meeting
```

```
Enter task priority: 7
```

```
Task added.
```

```
Choose an option:
```

1. Add a task

Enter task description: Attend Meeting

Enter task priority: 7

Task added.

Choose an option:

1. Add a task
2. Remove the highest priority task
3. Update a task's description
4. Display tasks
5. Exit

4

Priority: 7, Description: Attend Meeting

Priority: 4, Description: Buy Fire Crackers

Priority: 3, Description: Buy Milk

Choose an option:

1. Add a task
2. Remove the highest priority task
3. Update a task's description
4. Display tasks
5. Exit

3

Updating an item's description:

Enter the priority of the task you want to update:

3

Enter the new task description: Buy Milk and Eggs

Priority: 7, Description: Attend Meeting

Priority: 4, Description: Buy Fire Crackers

Priority: 3, Description: Buy Milk and Eggs

Choose an option:

1. Add a task
2. Remove the highest priority task
3. Update a task's description
4. Display tasks
5. Exit

Updating an item's description:
Enter the priority of the task you want to update:

3

Enter the new task description: Buy Milk and Eggs

Priority: 7, Description: Attend Meeting

Priority: 4, Description: Buy Fire Crackers

Priority: 3, Description: Buy Milk and Eggs

Choose an option:

1. Add a task
2. Remove the highest priority task
3. Update a task's description
4. Display tasks
5. Exit

2

Removing the highest priority item:

Choose an option:

1. Add a task
2. Remove the highest priority task
3. Update a task's description
4. Display tasks
5. Exit

4

Priority: 4, Description: Buy Fire Crackers

Priority: 3, Description: Buy Milk and Eggs

Choose an option:

1. Add a task
2. Remove the highest priority task
3. Update a task's description
4. Display tasks
5. Exit

5

Exiting the program.

PS E:\PG - DAC\DSA\Mini Project> █