Task Priority

DFS Lab Project report by -1901,1903,1921

Team Members:

1901-Aishwarya Ganesh

1903-Meliva Cruz

1921-Shamly Kotkar

Problem Statement :

Implementation of Priority Queues using Linked Lists and Arrays

Detailed Project Description:

The concept of Tasks and Task of highest priority was used for this project. Task that have the highest priority are completed first and if there is are two or more tasks with the same priority the task that was added 1st is completed 1st. The higher numbers denote higher priority and lower numbers denote lower. Priority can only be in integers. The only way to complete a task is if it is the highest priority task i.e in front of the queue. Our project also allows the user to add tasks, display tasks, show list of all the higher priority tasks, number of tasks in the task list and modify the task name.

Priority Queue using Linked List:

Since we used a linked list there was no restriction for number of tasks and thus this meant we did not have to set an upper or lower bound for priority so users could set it as they wish which meant they were not restricted to a set of priorities. The tasks were added in to the queue in sorted descending order of priority.

Priority Queue using Array:

Since we also used an array to implement the priority queue, there was restriction for number of tasks that could be inserted into the array, as the array size is fixed. We cannot utilize the blank spaces as the front pointer moves ahead. The tasks were added into the queue in sorted descending order of their priority.

Details of the Data Structures used:

Linked List

we used a struct data type called TaskNode which consisted of:

- char data[20] : stores name of task
- int priority: stores the priority value of task
- TaskNode* Link: which stored address of next node/task

Array

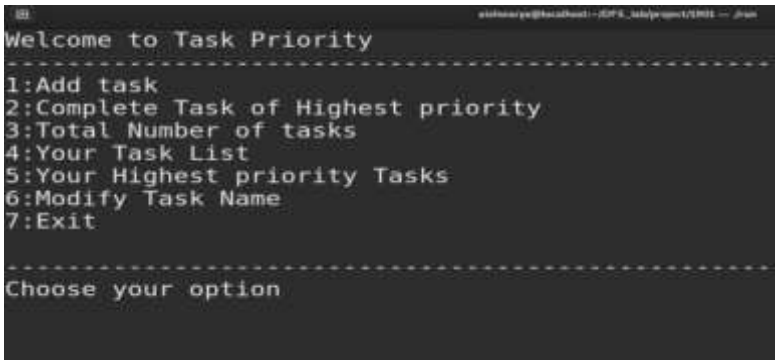
we used a struct data type called queue which consisted of

- char data[10] : stores name of task
- int priority: stores the priority value of task

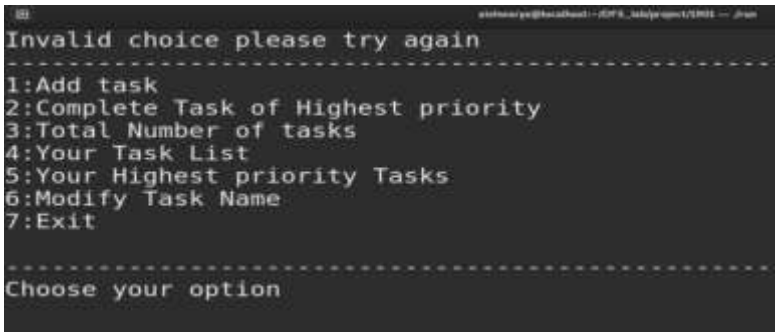
Screen shots:

Priority Queue using Linked List:

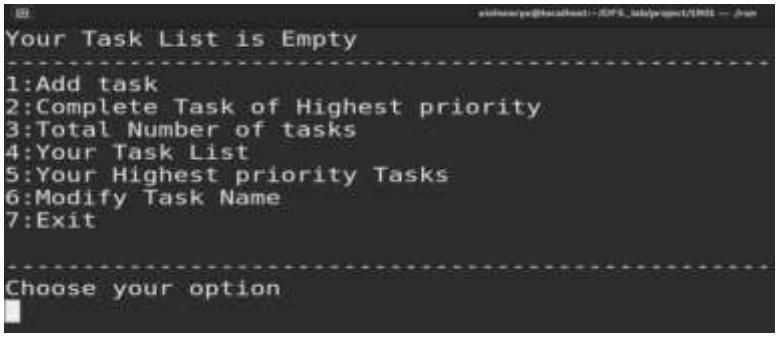
Main menu:



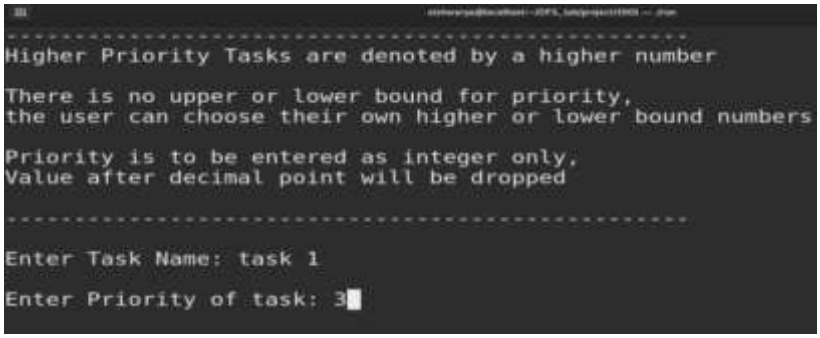
Main Menu Invalid Choice:



Empty Task List:

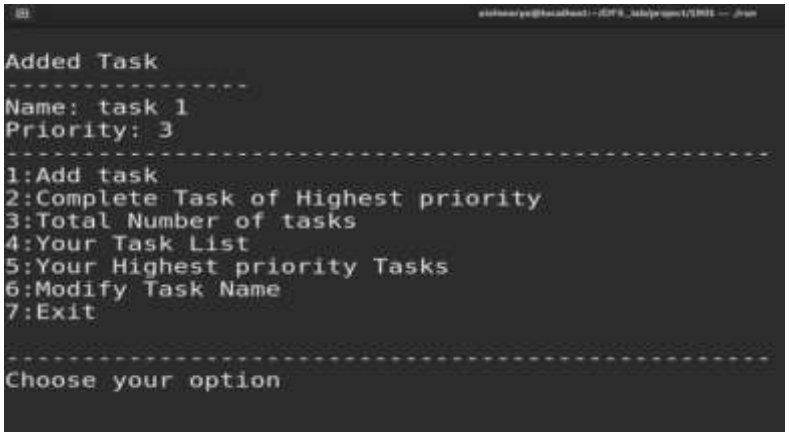


Add Task:

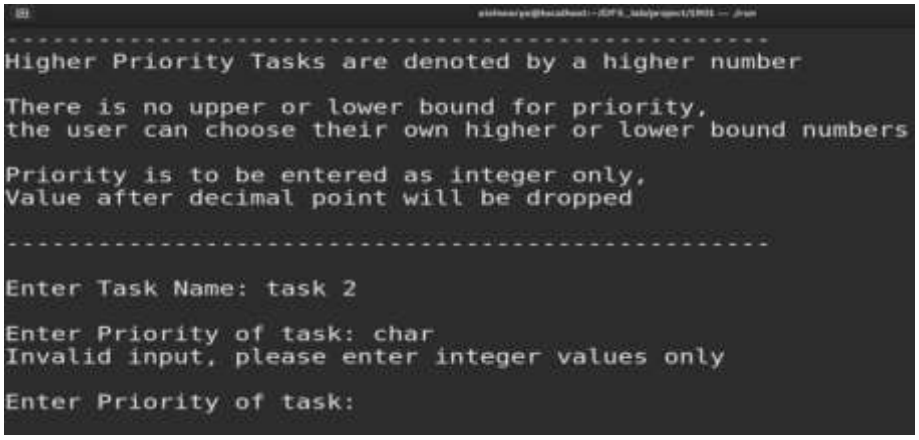


Task Add Confirmation:

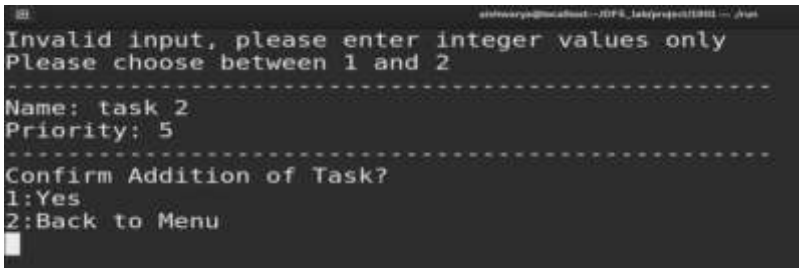
Task Added:



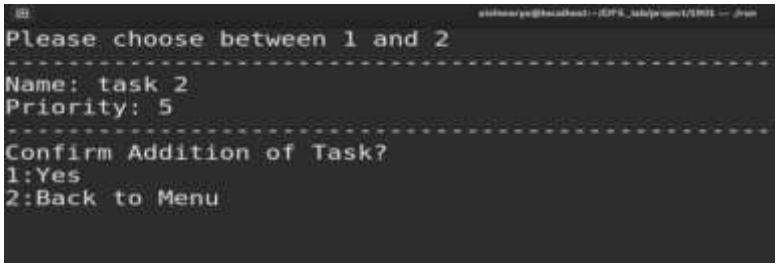
Add Task Invalid Priority Input Response:



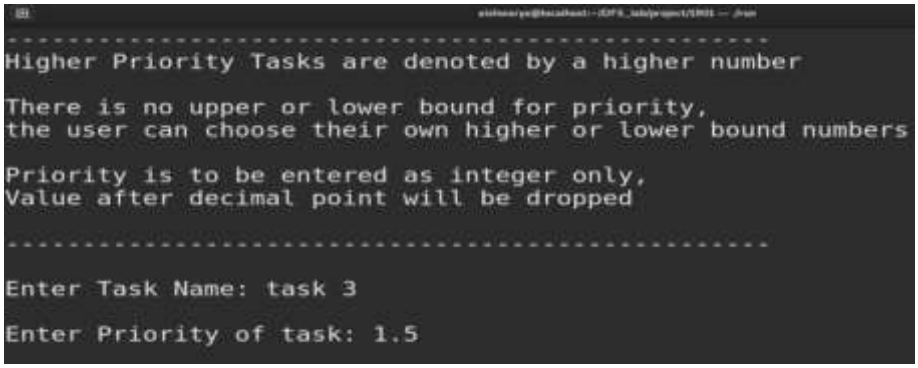
Task Add Confirmation mixed string response:

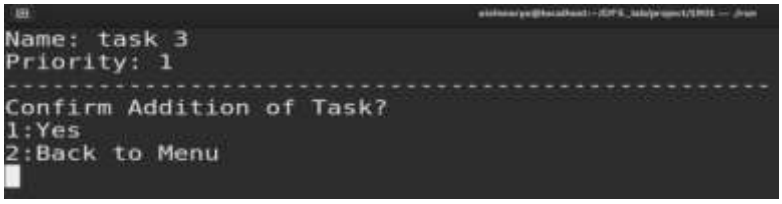


Task Add Confirmation Invalid Integer Response:

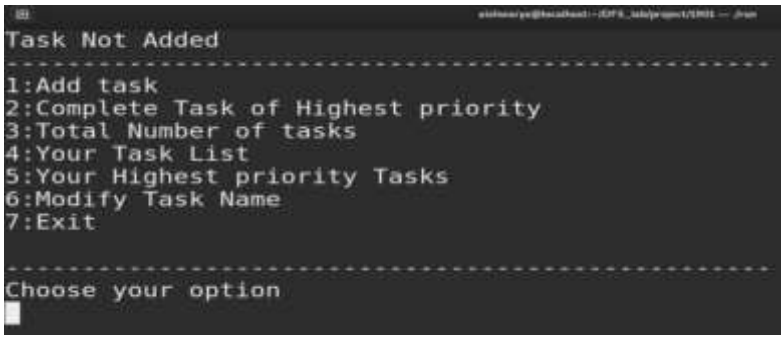


Task Add Float Priority Value:

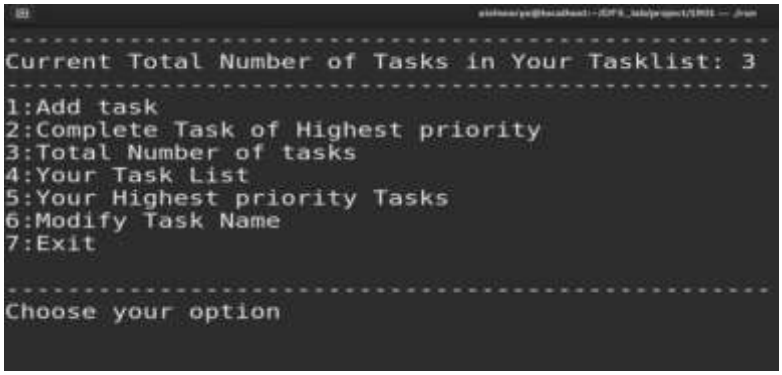




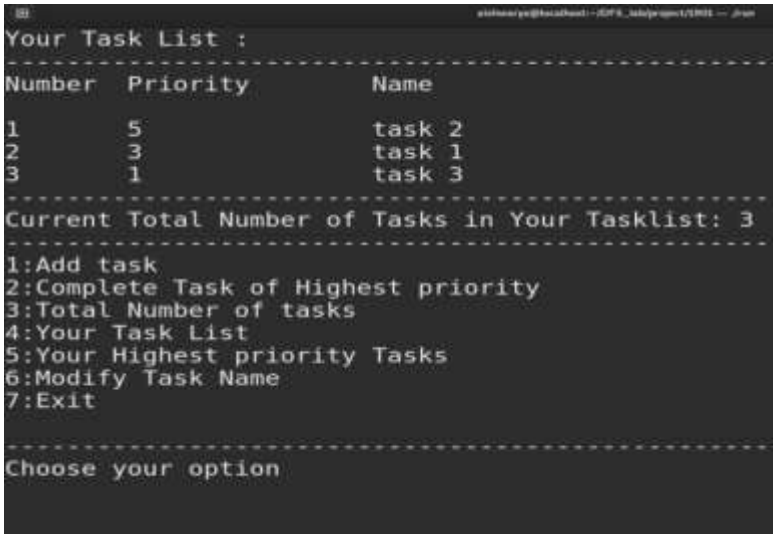
Task Add Cancelled:



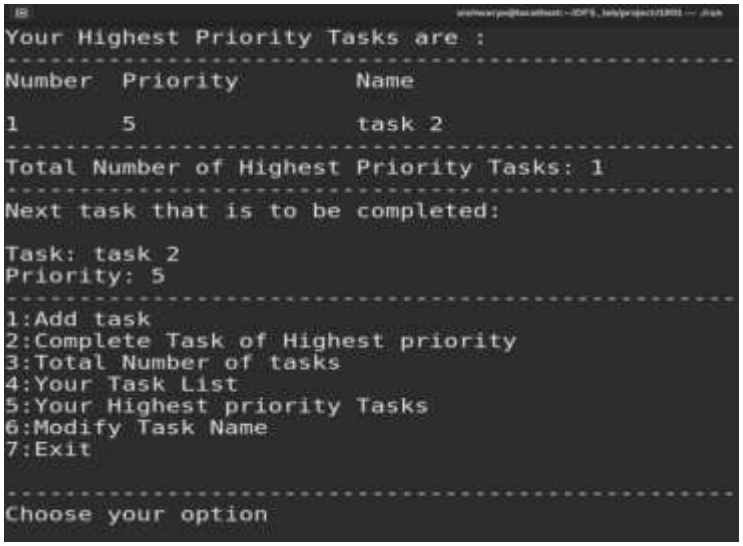
Total number of Tasks:



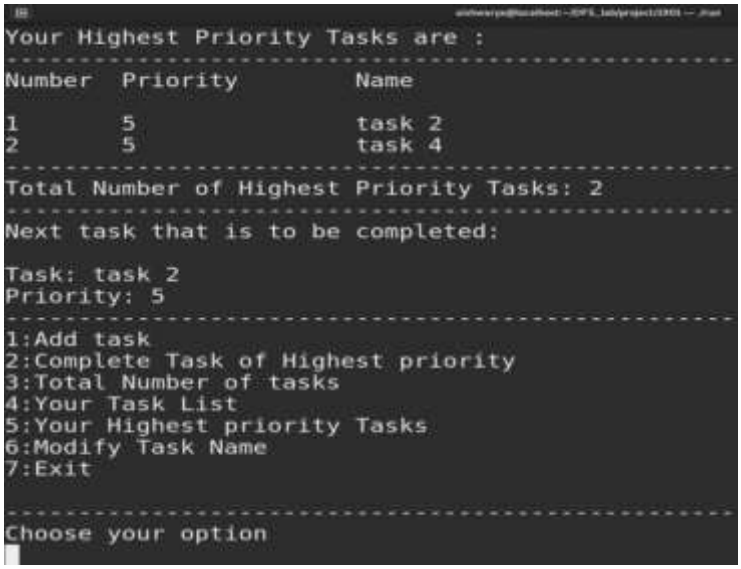
Your Task List:



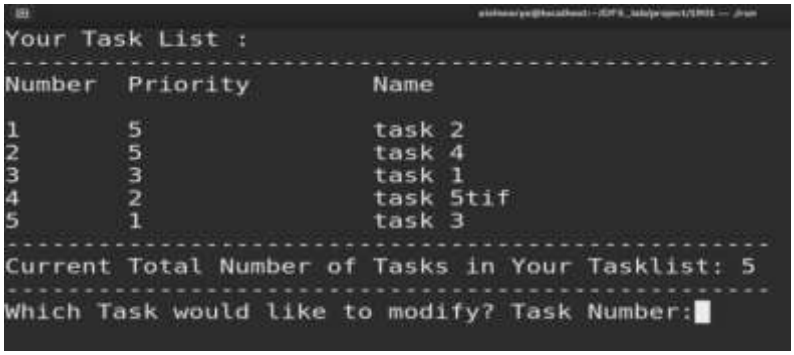
Highest Priority Tasks:
If only 1 Highest Priority task:



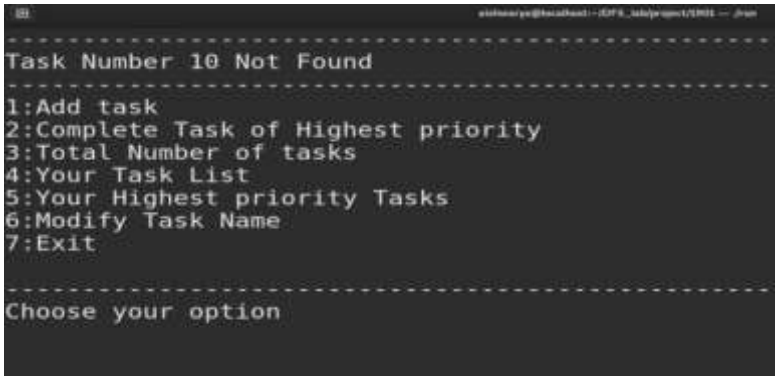
If more than one task has highest priority Task Added 1st is to be completed first:



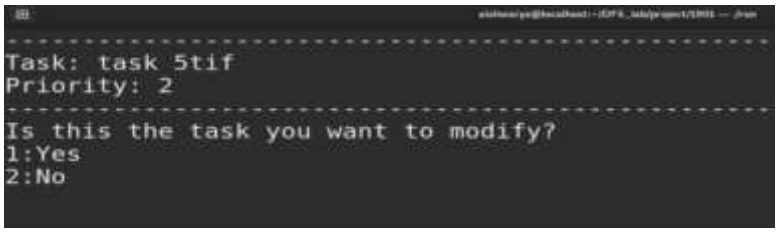
Modify Task:



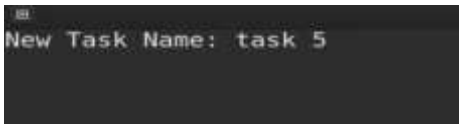
Task Number Not in list:

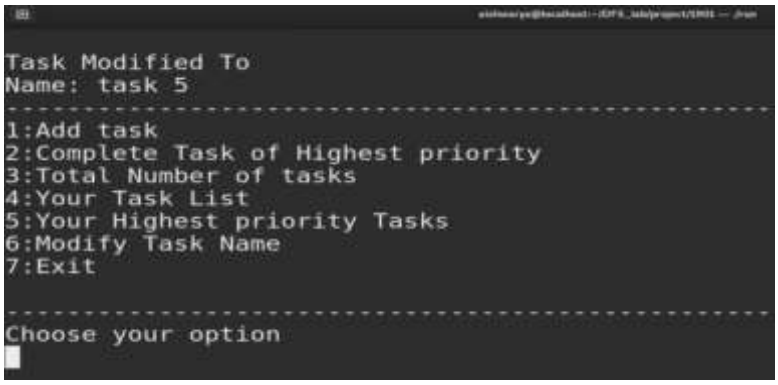


Task Modification Confirmation:

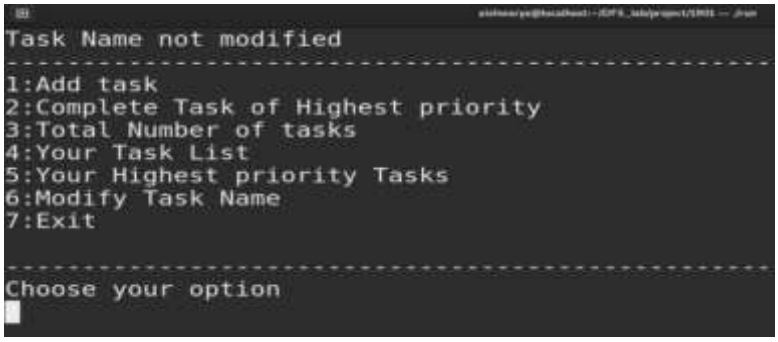


Task Name Modification Confirmed:

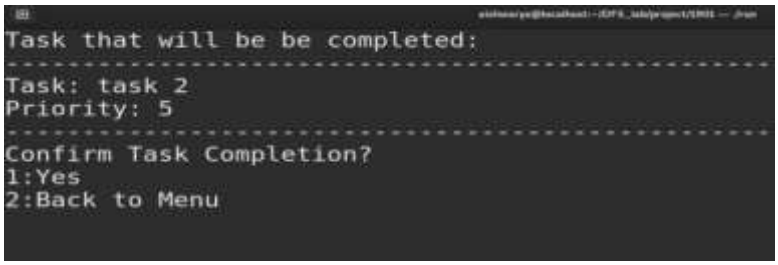




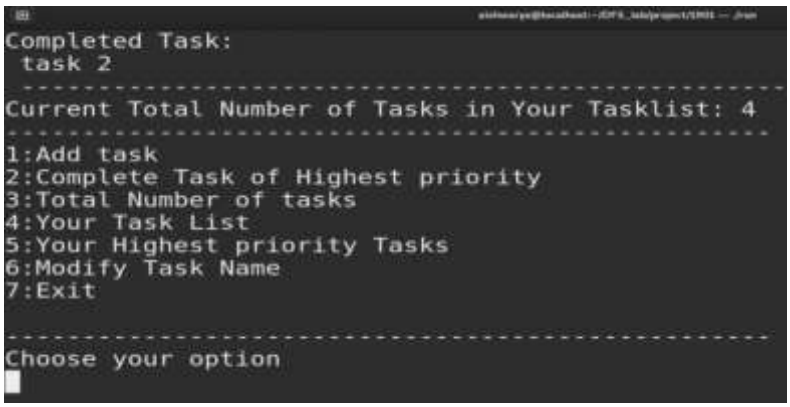
Task Name Modification Cancelled:



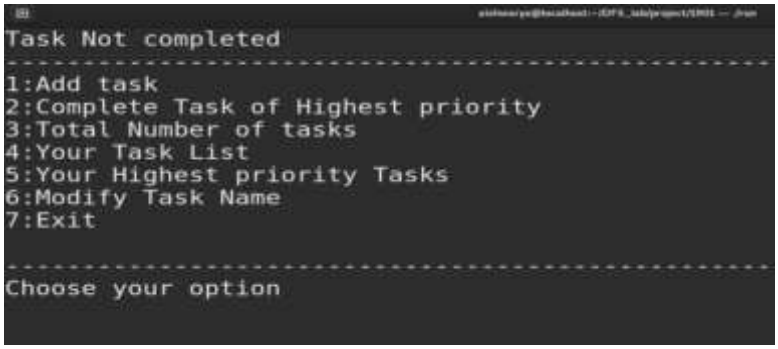
Task Completion:



Task Completion Confirmed:



Task Completion Cancelled:

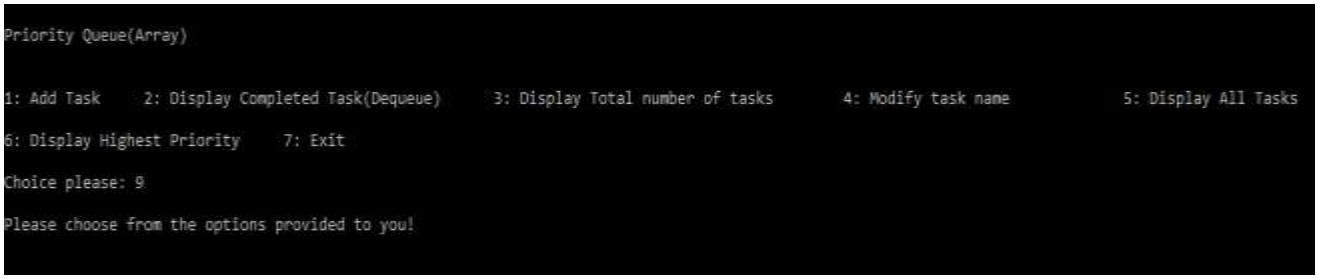


Exit Screen:

Priority Queue using Array:

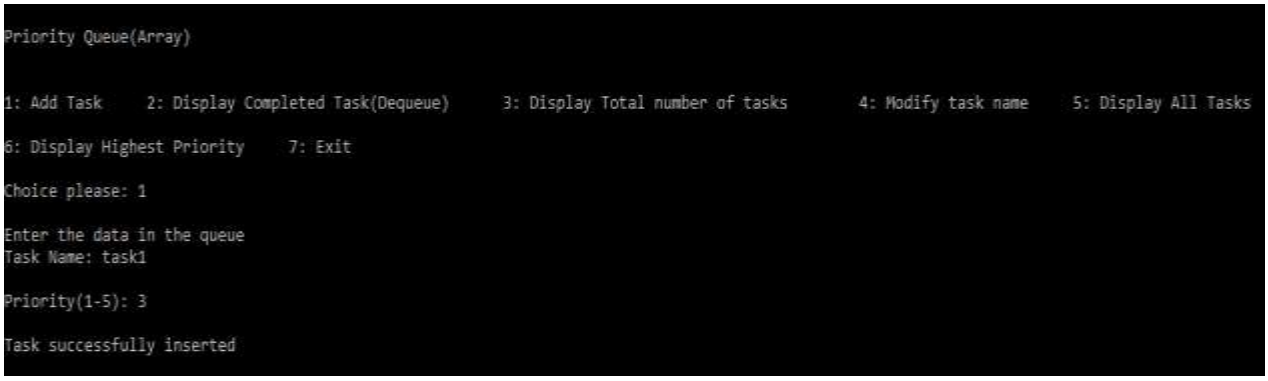
Main Menu

Main Menu Invalid Choice

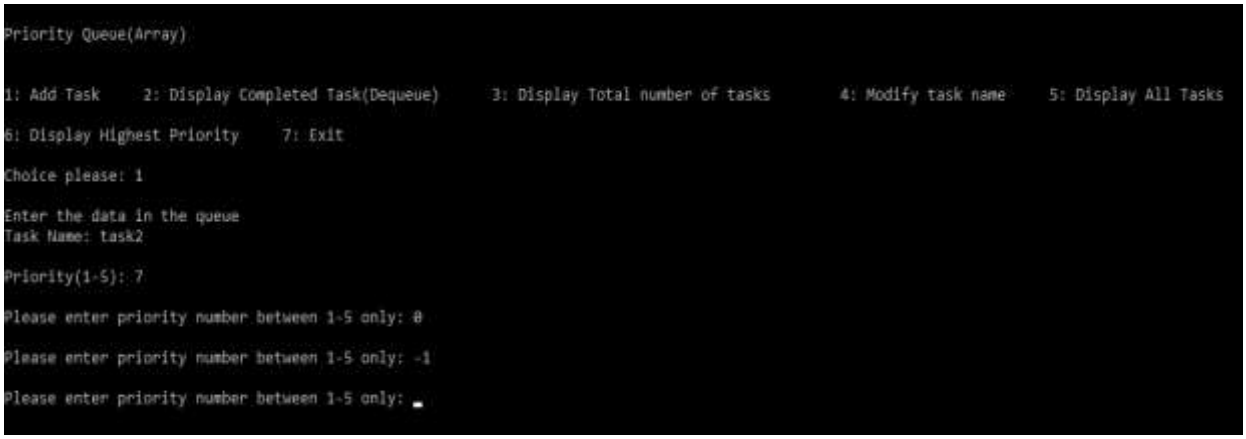


Enqueue

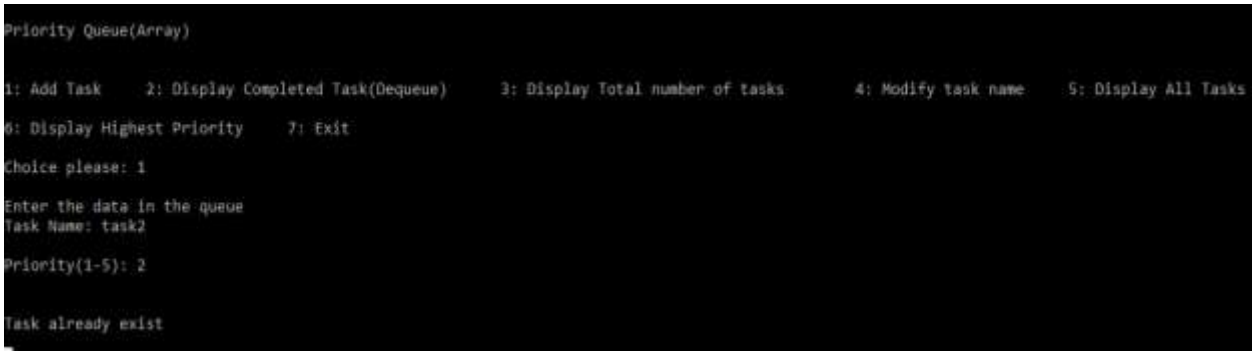
Valid Entries



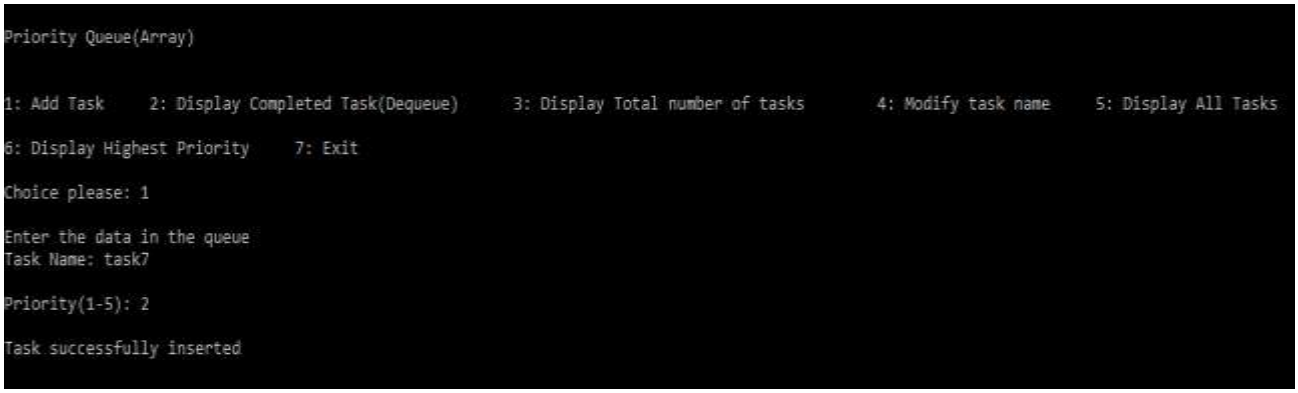
Invalid Priority Value



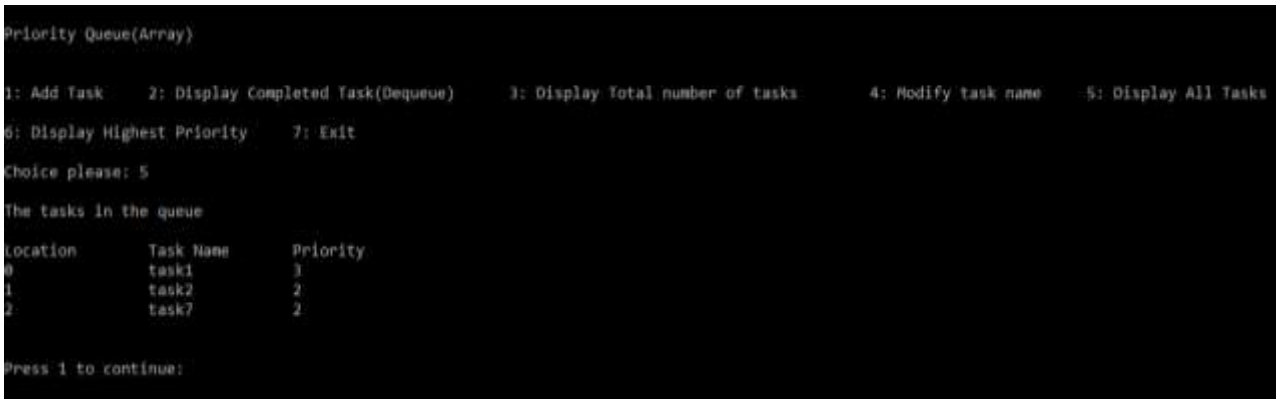
Duplicate Entry



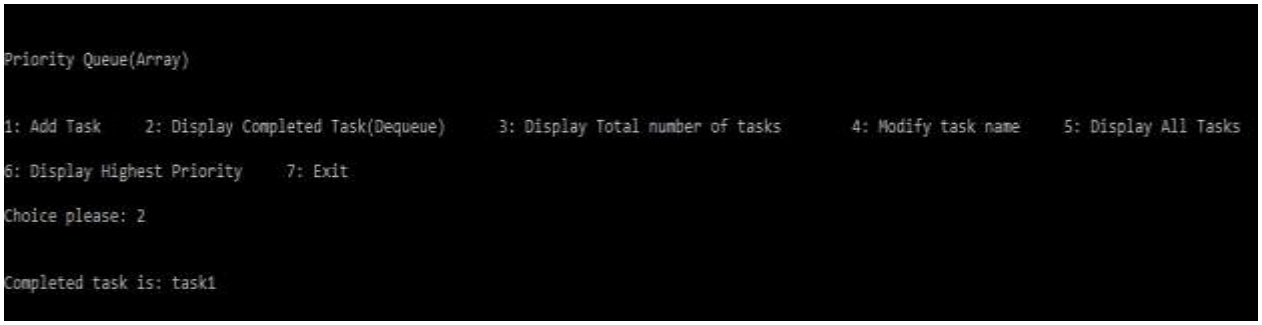
Same Priority but different Task Name

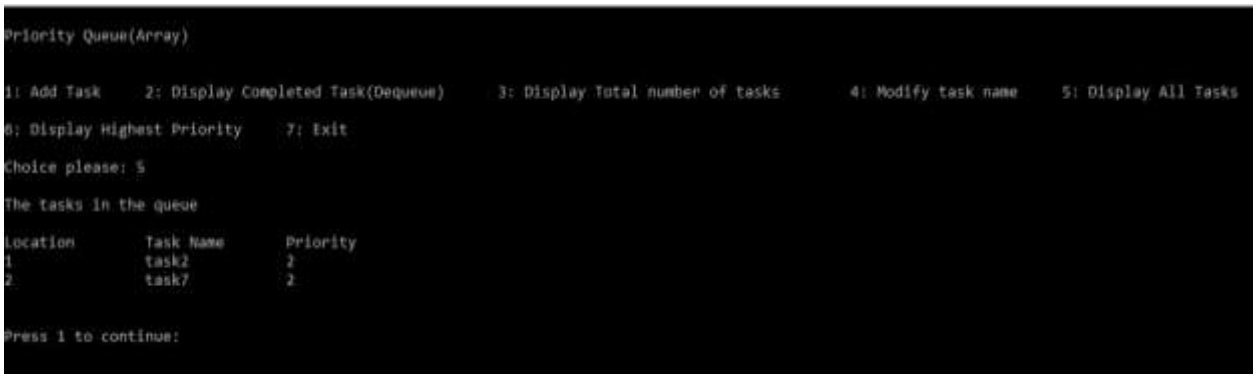


Display



Dequeue / Display completed task





Display Total Number of Tasks

```
Priority Queue(Array)

1: Add Task     2: Display Completed Task(Dequeue)      3: Display Total number of tasks        4: Modify task name     5: Display All Tasks

6: Display Highest Priority     7: Exit

Choice please: 3

Total Number of tasks in the queue: 2
```

Modify

Valid Input

```
Priority Queue(Array)

1: Add Task     2: Display Completed Task(Dequeue)      3: Display Total number of tasks        4: Modify task name     5: Display All Tasks

6: Display Highest Priority     7: Exit

Choice please: 4

The tasks in the queue

Location        Task Name       Priority
1               task1           3
2               task2           2
3               task7           2

Enter name of the task you wanna modify: task7

Priority(1-5): 2

Enter new name: task9

Task name modified successfully
```

```
Priority Queue(Array)

1: Add Task     2: Display Completed Task(Dequeue)      3: Display Total number of tasks        4: Modify task name     5: Display All Tasks

6: Display Highest Priority     7: Exit

Choice please: 5

The tasks in the queue

Location        Task Name       Priority
1               task1           3
2               task2           2
3               task9           2

Press 1 to continue:
```

Invalid Input

```
Priority Queue(Array)

1: Add Task     2: Display Completed Task(Dequeue)      3: Display Total number of tasks        4: Modify task name     5: Display All Tasks

6: Display Highest Priority     7: Exit

Choice please: 4

The tasks in the queue

Location        Task Name       Priority
1               task1           3
2               task2           2
3               task9           2

Enter name of the task you wanna modify: task1

Priority(1-5): 2

Couldn't find task named task1 with priority 2
```

Display Highest Priority

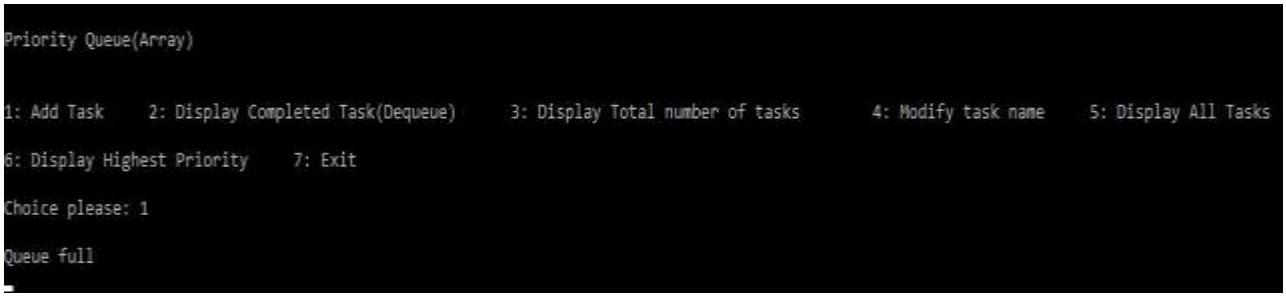
```
Priority Queue(Array)

1: Add Task     2: Display Completed Task(Dequeue)      3: Display Total number of tasks        4: Modify task name     5: Display All Tasks

6: Display Highest Priority     7: Exit

Choice please: 6

Highest Priority task
Task Name: task1        Priority: 3
```

Empty Queue

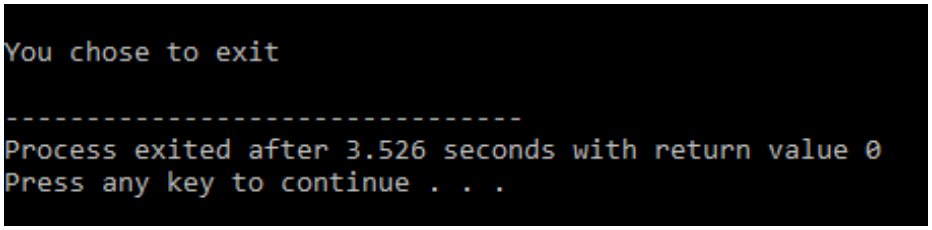
```
Priority Queue(Array)

1: Add Task     2: Display Completed Task(Dequeue)      3: Display Total number of tasks        4: Modify task name     5: Display All Tasks

6: Display Highest Priority     7: Exit

Choice please: 4

Its an empty queue
```

```
Priority Queue(Array)

1: Add Task     2: Display Completed Task(Dequeue)      3: Display Total number of tasks        4: Modify task name     5: Display All Tasks

6: Display Highest Priority     7: Exit

Choice please: 3

No Tasks in the queue
```

Queue Full



Exit



Division of work:

Implementation using Linked List: 1901

Implementation using Arrays: 1903,1921

Any interesting problems faced and their solutions:

1. Scanning of integers and then scanning string using scanf would cause the string input to get skipped to get rid of this and clear the buffer of non int characters the statement:

   while((getchar())!= '\n');

   was used to clear the buffer;

2. This was also a part of a solution that was used for the data validation of integers, if non numerical characters were input

   The following statements were used to check if only integers were input:

   ```
   status = scanf("%d",&ch);     scanf returns 1 on valid datatype input and 0 on invalid
   while((getchar())!= '\n');    //clear input buffer of non int chars

   if(status!=1)//no integer is input
       printf("Invalid input, please enter integer values only\n");
   ```

   scanf returns 1 on valid datatype input and 0 on invalid and here pure numerical character input is valid and mixed string is invalid. Thus storing the return value and checking it allowed us to tell the user if they didn't input the expected integers.