**CODE:-**

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define MAX 5

typedef struct

{

int TaskId;

char TaskTitle[20];

int TaskDuration;

char Status;

} TASK;

int front = -1, rear = -1;

void insert(int[], int);

int delete (int[]);

int isFull()

{

if (rear == MAX - 1)

return 1;

else

return 0;

}

int isEmpty()

{

if (front == -1 || front == rear + 1)

return 1;

else

return 0;

}

void insert(int a[], int item)

{

if (front == -1)

front = 0;

rear = rear + 1;

a[rear] = item;

}

int delete (int a[])

{

int item;

if (isEmpty())

{

printf("Queue Underflow\n");

exit(1);

}

item = a[front];

for (int i = 1; i < MAX; i++)

a[i - 1] = a[i];

rear--;

return item;

}

void display(int a[], TASK T[])

{

printf("The details of queued tasks are :\n");

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

{

for (int j = 0; j < 10; j++)

{

if (a[i] == T[j].TaskId)

{

printf("TaskId: %d\nTaskTitle: %s\nTaskDuration: %d\nStatus: %c\n", T[j].TaskId, T[j].TaskTitle, T[j].TaskDuration, T[j].Status);

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

}

}

}

}

void delay(int t)

{

long int t3;

clock\_t t1, t2;

t3 = t \* CLOCKS\_PER\_SEC;

t1 = t2 = clock();

while ((t2 - t1) < t3)

t2 = clock();

}

int main()

{

TASK T[10];

FILE \*fp;

int Q[MAX];

fp = fopen("task.txt", "r");

for (int i = 0; i < 10; i++)

{

fscanf(fp, "%d%s%d %c", &T[i].TaskId, T[i].TaskTitle, &T[i].TaskDuration, &T[i].Status);

}

while (1)

{

int ch, id, num;

printf("Enter 1 to schedule a task.\n");

printf("Enter 2 to run the task at start of queue.\n");

printf("Enter 3 to display details of tasks that are queued up.\n");

printf("Enter 4 to exit.\n");

scanf("%d", &ch);

switch (ch)

{

case 1:

printf("Enter the task ID.\n");

scanf("%d", &id);

for (int i = 0; i < 10; i++)

{

if (T[i].TaskId == id)

{

if (T[i].Status == 'C' || T[i].Status == 'Q')

printf("Task cannot be scheduled.\n");

else if (!isFull())

{

insert(Q, T[i].TaskId);

T[i].Status = 'Q';

}

else

{

int time1, time2 = 0;

for (int j = rear; j >= front; j--)

{

for (int k = 0; k < 10; k++)

{

if (Q[j] == T[k].TaskId)

{

time1 = T[k].TaskDuration;

}

}

for (int k = 0; k < 10; k++)

{

if (Q[j] == T[k].TaskId)

{

time2 += T[k].TaskDuration;

}

}

}

printf("The minimum and maximum waiting times are %d seconds and %d seconds respectively.\n", time1, time2);

}

}

}

break;

case 2:

num = delete (Q);

for (int i = 0; i < 10; i++)

{

if (T[i].TaskId == num)

{

delay(T[i].TaskDuration);

T[i].Status = 'C';

}

}

break;

case 3:

display(Q, T);

break;

case 4:

exit(1);

default:

printf("Erroneous input.\n");

}

}

fclose(fp);

return 0;

}

**OUTPUT:-**

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

1

Enter the task ID.

100

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

1

Enter the task ID.

200

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

1

Enter the task ID.

100

Task cannot be scheduled.

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

1

Enter the task ID.

300

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

1

Enter the task ID.

400

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

1

Enter the task ID.

500

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

1

Enter the task ID.

550

The minimum and maximum waiting times are 5 seconds and 26 seconds respectively.

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

3

The details of queued tasks are :

TaskId: 100

TaskTitle: New1

TaskDuration: 5

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TaskId: 200

TaskTitle: New2

TaskDuration: 5

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TaskId: 300

TaskTitle: New3

TaskDuration: 5

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TaskId: 400

TaskTitle: New4

TaskDuration: 5

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TaskId: 500

TaskTitle: New5

TaskDuration: 6

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

2

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

3

The details of queued tasks are :

TaskId: 200

TaskTitle: New2

TaskDuration: 5

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TaskId: 300

TaskTitle: New3

TaskDuration: 5

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TaskId: 400

TaskTitle: New4

TaskDuration: 5

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TaskId: 500

TaskTitle: New5

TaskDuration: 6

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

1

Enter the task ID.

550

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

3

The details of queued tasks are :

TaskId: 200

TaskTitle: New2

TaskDuration: 5

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TaskId: 300

TaskTitle: New3

TaskDuration: 5

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TaskId: 400

TaskTitle: New4

TaskDuration: 5

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TaskId: 500

TaskTitle: New5

TaskDuration: 6

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TaskId: 550

TaskTitle: New6

TaskDuration: 6

Status: Q

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter 1 to schedule a task.

Enter 2 to run the task at start of queue.

Enter 3 to display details of tasks that are queued up.

Enter 4 to exit.

4