main.c

```c
/*Consider there are 5 Tasks T1, T2, T3, T4 and T5. Task1 takes 50 seconds to execute,
similarly Task2 40 sec, Task3 30 sec, Task4 20 sec Task5 10 sec to complete the execution
and the condition is each task is given the fixed timestamp of 10 seconds that they can get the
CPU for only 10 seconds after which the other task has to get the CPU for their execution.
If Task1 has got the CPU for 10 seconds and 40 seconds is left to complete its execution it
has to join at the back of Task5. Similarly, for other tasks which has not completed has to
join at the back of other tasks for its turn to get the CPU for execution. Identify the suitable
data structure and develop a application (C Program) to demonstrate above execution
system in the fair fashion.*/
#include<stdio.h>
#include<stdlib.h>
#define size 5
int front=0,rear=-1,q[size],count=0;
void insertrear(int item)
{
    if(count==size)
    {
        printf("queue overflow");
        return;
    }
    rear=(rear+1)%size;
    q[rear]=item;
    count++;
}
int deletefront()
{
    int item;
    if(count==0) return -1;
    item = q[front];
    front=(front+1)%size;
    count=count-1;
    return item;
}
void displayq()
{
    int i,f;
    if(count==0)
    {
        printf("\nqueue is empty");
```

main.c

```
    {
        printf("\nqueue is empty");
        return;
    }
    f=front;
    int c=count;
    printf("\nProgram in the queue");
    while(c!=0)
    {
        printf("\nExecution time : %d",q[f]);
        f=((f+1)%size);
        c--;
    }
}
void execution()
{
    if(count==0)
    {
        printf("\nqueue is empty");
        return;
    }
    int a,flag=0;
    while(flag<5)
    {
        a=deletefront();
        if(a!=0)
        {
            a-=10;
            flag=0;
        }
        insertrear(a);
        if(a==0)
        {
            flag++;
        }
    }
    printf("\nExecution completed");
    displayq();
    front=0,rear=-1;
}
```

main.c

```
        insertrear(a);
        if(a==0)
        {
            flag++;
        }
    }
    printf("\nExecution completed");
    displayq();
    front=0,rear=-1;
}
void main()
{
    int choice,item;
    for(;;)
    {
        printf("\n1.Insert rear\n2.Delete front\n3.Display\n4.Execution\n5.Exit\n");
        printf("Enter the choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("Enter the execution time of the program :");
            scanf("%d",&item);
            insertrear(item);
            break;
            case 2:item=deletefront();
            if(item==-1)
            printf("queue is empty\n");
            else
            printf("item deleted is %d \n",item);
            break;
            case 3:displayq();
            break;
            case 4:execution();
            break;
            default:exit(0);
        }
    }
}
```

input

```
1.Insert rear
2.Delete front
3.Display
4.Execution
5.Exit
Enter the choice : 1
Enter the execution time of the program :50

1.Insert rear
2.Delete front
3.Display
4.Execution
5.Exit
Enter the choice : 1
Enter the execution time of the program :40

1.Insert rear
2.Delete front
3.Display
4.Execution
5.Exit
Enter the choice : 1
Enter the execution time of the program :30

1.Insert rear
2.Delete front
3.Display
4.Execution
5.Exit
Enter the choice : 1
Enter the execution time of the program :20

1.Insert rear
2.Delete front
3.Display
4.Execution
```

input

```
4.Execution
5.Exit
Enter the choice : 1
Enter the execution time of the program :10

1.Insert rear
2.Delete front
3.Display
4.Execution
5.Exit
Enter the choice : 3

Program in the queue
Execution time : 50
Execution time : 40
Execution time : 30
Execution time : 20
Execution time : 10
1.Insert rear
2.Delete front
3.Display
4.Execution
5.Exit
Enter the choice : 4

Execution completed
Program in the queue
Execution time : 0
Execution time : 0
Execution time : 0
Execution time : 0
Execution time : 0
1.Insert rear
2.Delete front
3.Display
4.Execution
5.Exit
```

main.c

```c
/*Consider the below scenario of linear queue. Design and implement a queue which utilize the
free space without shifting the elements of queue.*/

#include<stdio.h>
#include<stdlib.h>
#define size 4
int item,front=0,rear=-1,q[size],count=0;
void insertrear()
{
    if(count==size)
    {
        printf("queue overflow\n");
        return;
    }
    rear=(rear+1)%size;
    q[rear]=item;
    count++;
}
int deletefront()
{
    if(count==0) return -1;
    item=q[front];
    front=(front+1)%size;
    count=count-1;
    return item;
}
void displayQ()
{
    int i,f;
    if(count==0)
    {
        printf("queue is empty\n");
        return;
    }
    f=front;
    printf("Contents of queue \n");
    for(i=1;i<=count;i++)
    {
        printf("%d\n",q[f]);
```

main.c

```c
    }
    f=front;
    printf("Contents of queue \n");
    for(i=1;i<=count;i++)
    {
        printf("%d\n",q[f]);
        f=(f+1)%size;
    }
}
void main()
{
    int choice;
    for(;;)
    {
        printf("\n1:insertrear\n2:deletefront\n3:display\n4:exit\n");
        printf("enter the choice\n");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:printf("enter the item to be inserted\n");
            scanf("%d",&item);
            insertrear();
            break;
            case 2:item=deletefront();
            if(item==-1)
            printf("queue is empty\n");
            else
            printf("item deleted =%d\n",item);
            break;
            case 3:displayQ();
            break;
            case 4:exit(0);
            break;
            default:printf("Invalid choice\n");
        }
    }
}

```

input

```
1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
23

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
45

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
67

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
89
```

input

```
4:exit
enter the choice
1
enter the item to be inserted
90
queue overflow

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
2
item deleted =23

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
2
item deleted =45

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
3
Contents of queue
67
89

1:insertrear
2:deletefront
3:display
4:exit
```

input

```
4:exit
enter the choice
3
Contents of queue
67
89

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
44

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
3
Contents of queue
67
89
44

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
4


...Program finished with exit code 0
Press ENTER to exit console.
```