

NAME: Deepanshu Gupta

SEC: AI & ML

CLASS ROLL NO.: 10

PROBLEM STATEMENT: Write a C program for implementing process scheduling for a CPU in multiprogramming environment in a time changing basic. Input no. of process, time taken by every process and CPU slot.

CODE:

```
#include <stdio.h>

#include <stdlib.h>

typedef struct node
{
    struct node *next;
    int info;
    int sno;
} NODE;

NODE* insert(NODE *start,int n)
{
    static int sno=0;
    NODE *p=NULL;
    p=(NODE*)malloc(sizeof(NODE));
    if(p!=NULL)
    {
        p->sno=++sno;
        p->info=n;
        if(start==NULL)
            p->next=p;
        else
        {
            p->next=start->next;
```

```

        start->next=p;
    }
    start=p;
    return start;
}
}

void display(NODE *start)
{
    if (start==NULL)
        printf("List is empty\n");
    else
    {
        NODE *end=start;
        printf("Time Taken\n");
        while (start->next!=end)
        {
            printf("%d\n", start->info);
            start=start->next;
        }
        printf("%d\n",start->info);
    }
}

void delete(NODE **p)
{
    NODE *q=*p , *r=NULL;
    if(q->next==q)
    {
        free(q);
        *p=NULL;
    }
}

```

```

else
{
    r=q->next;
    q->next=r->next;
    free(r);
    *p=q;
}
}

void taskprocess(NODE **p)
{
    int timee, c=0;
    printf("Enter time to allocate:");
    scanf("%d", &timee);
    NODE *q = *p;
    while(q!=NULL)
    {
        c++;
        int y= c*timee;
        NODE *f = q->next;
        f->info = (f->info)-timee;
        if((f->info)<=0)
        {
            printf("Process-%d is completed in %d turn within %d unit time\n", f->sno,c,y);
            delete(&q);
        }
        else
            q=q->next;
    }
}

int main()

```

```

{
    NODE *top=NULL;

    int choice, x;

    do
    {
        printf("OPERATION YOU NEED TO PERFORM : \n1.) INSERT\n2.) PROCESS
SCHEDULING\n3.) DISPLAY\n4.) EXIT\nEnter your choice: ");

        scanf("%d",&choice);

        switch(choice)
        {
            case 1:
            {
                printf("Enter the number to be inserted : ");

                scanf("%d",&x);

                top=insert(top, x);

                break;
            }
            case 2:
            {
                taskprocess(&top);

                exit(0);
            }
            case 3:
            {
                display(top);

                break;
            }
            case 4:
            {
                exit(0);
            }
        }
    }
}

```

```
default:
{
    printf("Invalid choice hence exit\n");
    break;
}
}
} while(choice<=4);
}
```

OUTPUT:

OPERATION YOU NEED TO PERFORM :

- 1.) INSERT
- 2.) PROCESS SCHEDULING
- 3.) DISPLAY
- 4.) EXIT

Enter your choice: 1

Enter the number to be inserted : 10

OPERATION YOU NEED TO PERFORM :

- 1.) INSERT
- 2.) PROCESS SCHEDULING
- 3.) DISPLAY
- 4.) EXIT

Enter your choice: 1

Enter the number to be inserted : 10

OPERATION YOU NEED TO PERFORM :

- 1.) INSERT
- 2.) PROCESS SCHEDULING
- 3.) DISPLAY
- 4.) EXIT

Enter your choice: 1

Enter the number to be inserted : 30

OPERATION YOU NEED TO PERFORM :

- 1.) INSERT
- 2.) PROCESS SCHEDULING
- 3.) DISPLAY
- 4.) EXIT

Enter your choice: 1

Enter the number to be inserted : 40

OPERATION YOU NEED TO PERFORM :

- 1.) INSERT
- 2.) PROCESS SCHEDULING
- 3.) DISPLAY
- 4.) EXIT

Enter your choice: 3

Time Taken

40

10

10

30

OPERATION YOU NEED TO PERFORM :

- 1.) INSERT
- 2.) PROCESS SCHEDULING
- 3.) DISPLAY
- 4.) EXIT

Enter your choice: 2

Enter time to allocate:10

Process-1 is completed in 1 turn within 10 unit time

Process-2 is completed in 2 turn within 20 unit time

Process-3 is completed in 7 turn within 70 unit time

Process-4 is completed in 9 turn within 90 unit time