

Name - Anshika Rana RollNo - 2023033 Campus - D.Dun
 Course - Bsc. IT'A' Student Id - 20052010 Sem - 2.
 Subject - Operating System lab Subject code -

Ques -

```
#include<stdio.h>
unsigned int Heap[10000], Index[10000], Position[10000];
size = 0;
unsigned int Temp[10000], TempI[10000];
unsigned int Arr_Time[10000], Cook_Time[10000], Num;
void merge(int Low, int Mid, int High)
{
    int i = Low, j = Mid + 1, k = 0;
    while (i <= Mid && j <= High)
    {
        if (Arr_Time[i] <= Arr_Time[j])
        {
            Temp[k] = Arr_Time[i];
            TempI[k] = Cook_Time[i];
            i++;
            k++;
        }
        else
        {
            Temp[k] = Arr_Time[j];
            TempI[k] = Cook_Time[j];
            j++;
        }
    }
}
```

Teacher's Signature

Anshika Rana
22 June 21

```

j++;
k++;
}
if (i <= Mid)
{
    int I;
    for (I = i; I <= Mid; I++)
    {
        Temp[k] = Arr_Time[I];
        Temp1[k] = Cook_Time[I];
        k++;
    }
    else if (j <= High)
    {
        int I;
        for (I = j; I <= High; I++)
        {
            Temp[k] = Arr_Time[I];
            Temp1[k] = Cook_Time[I];
            k++;
        }
    }
    K = 0;
    for (i = Low; i <= High; i++)
    {
        Arr_Time[i] = Temp[k];
        Cook_Time[i] = Temp1[k];
        k++;
    }
}

```

Anubha Rana

Expt. No. _____

Page. No. _____ Date _____

```
void divide (int low, int High)
{
```

```
    if (low < High)
    {
```

```
        int Mid = (low + High) / 2;
```

```
        divide (low, Mid);
```

```
        divide (Mid + 1, High);
```

```
        merge (low, Mid, High);
```

```
}
```

```
void Insert (int Node, unsigned int value)
```

```
int s;
```

```
if (Position [Node] == 0)
{
```

```
    Heap [++size] = value;
```

```
    Index [size] = Node;
```

```
    Position [Node] = size;
```

```
    s = size;
```

```
}
```

```
else
{
```

```
    Heap [Position [Node]] = value;
```

```
    s = Position [Node];
```

```
}
```

```
while (s != 1)
```

Teacher's Signature _____

Anilca Rane
22-June-21

```

    {
        if (Heap[S/2] > Heap[S])
        {
            int t = Heap[S/2];
            Heap[S/2] = Heap[S];
            Heap[S] = t;
            t = Index[S/2];
            Index[S/2] = Index[S];
            Index[S] = t;
        }
    }

```

```

    {
        else
        {
            break;
            S = S/2;
        }
    }

```

```

int Extract_Min()
{
    int M = Index[1];
    int S = 1;
}

```

```

Position[N] = -1;
Index[1] = Index[size];
Position[Index[size]] = 1;
Heap[1] = Heap[size--];
while(1)
{
}

```

Anushka Rane
 22-June-21

Expt. No.

Page. No.

Date

int T;

if (Heap[S*2] < Heap[S] && S*2 <= size || Heap[S*2+1] < Heap[S] && S*2+3)

{

if (Heap[S*2] < Heap[S*2+1])

T = S*2;

else

T = S*2 + 1;

int t = Heap[T];

Heap[T] = Heap[S];

Heap[S] = t;

t = Index[T];

Index[T] = Index[S];

Index[S] = t;

Position[Index[T]] = T;

Position[Index[S]] = S;

{

else

break;

S = T;

{

return N;

}

void Init(int N)

{

int i;

Teacher's Signature

Anubhav Agarwal
22 June 2021

```
for (i=0; i<=N; i++)
```

{

```
    position[i]=0;
```

```
    Index[i]=0;
```

```
    Heap[i]=1000000001;
```

```
    size=N;
```

{

```
int main()
```

{

```
int A-T, C-T, i=0;
```

```
long long wait-time=0, time=0;
```

```
scanf ("%d", &Num);
```

```
for (i=0; i<Num; i++)
```

```
scanf ("%Y.u%Y.u", &Arr-Time[i], &Cook-Time[i]);
```

```
divide(0, Num-1);
```

```
for (i=Num; i>=1; i-)
```

{

```
    Arr-Time[i]=Arr-Time[i-1];
```

```
    Cook-Time[i]=Cook-Time[i-1];
```

{

```
    Insert(i, Cook-Time[i]);
```

```
    i=2;
```

```
    while (i<=Num && Arr-Time[i]==Arr-Time[i])
```

{

```
        Insert(i, Cook-Time[i]);
```

```
        i++;
```

```
    while (size!=0)
```

{

```
        int T=Extract-Min();
```

```
        if (Time>Arr-Time[T])
```

Anubhav
Jain



{

(i+1) wait_time[i] = 0;

wait_time[i] += Time - Arr_Time[i] + Cook_Time[i];

Time[i] = Cook_Time[i];

}

else

{

Time[i] = Arr_Time[i] + Cook_Time[i];

wait_time[i] = Cook_Time[i];

}

T = i;

while (i <= Num && Arr_Time[i] <= Time)

{

o = emit, o = emit_time print print

(min, "ba")

}

(i+1) wait_time[i] = 0;

((T)oemit, T) wait_time["ba"] = 0;

((T)oemit, o) print

(-i, T < i, min = i)

((T)oemit, o) = ((T)oemit, o)

((T)oemit, o) = ((T)oemit, o)

((T)oemit, o) = ((T)oemit, o)

((T)oemit, o) = ((T)oemit, o) (88 min = si) print

((T)oemit, o), i) break

(o = 1) while Range

Ans hikka

21-June-21

Omit history 22-June-21