

Name - Abhinav Bhatt
University Roll No. - 2023013
Student I.d. - 20051037
Course - BSc I.T.
Section - B

PRACTICAL

Q1) Problem Statement \rightarrow Tieu owns a pizza restaurant and he manages it in his own way.

Ans) Code

```
#include <stdio.h>
unsigned int Heap[100001], Index[100001], Position[100001], Size=0;
unsigned int Temp[100001], Temp1[100001];
unsigned int Arr_Time[100001], Cook_Time[100001], 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];
            Temp1[k] = Cook_Time[i];
            i++;
            k++;
        }
        else
        {
            Temp[k] = Arr_Time[j];
            Temp1[k] = Cook_Time[j];
        }
    }
}
```

Date - 22/6/2021
Abhatt

```

    }++;
    }++;
}
}
if (i <= Mid)
{
    int I;
    for (I = i; I <= Mid; I++)
    { Temp[k] = Arr_Time[I]; Temp_1[k] = Cook_Time[I]; k++; }
}
else if (j <= High)
{
    int I;
    for (I = j; I <= High; I++)
    { Temp[k] = Arr_Time[I]; Temp_1[k] = Cook_Time[I]; k++; }
}
k = 0;
for (i = Low; i <= High; i++)
{
    Arr_Time[i] = Temp[k];
    Cook_Time[i] = Temp_1[k];
    k++;
}
}

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)
    {

```

```


```

```

        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;

```

```

            Position [Index [S/2]] = S/2;

```

```

            Position [Index [S/2]] = S;
        }
    }

```

Abhatt


```
else  
break;  
S = S/2;
```

```
}
```

```
}
```

Ashatt

Name - Abhinav Bhatt
University Roll No - 2023013
Student Id - 20051037
Course - BSc I.T.
Section - B

PRACTICAL

Q2) Write a C Program to implement SJF Algorithm and find out turn around time, waiting time, average turn around time and average waiting time.

Ans → Code

```
#include <stdio.h>
int main ()
{
    int bt[20], p[20], wt[20], tat[20], i, j, n, total=0, pos, temp;
    float avg_wt, avg_tat;
    printf ("Enter number of process :");
    scanf ("%d", &n);
    printf ("\nEnter Burst Time:\n");
    for (i=0; i<n; i++)
    {
        printf ("p%d:", i+1);
        scanf ("%d", &bt[i]);
        p[i] = i+1;
    }
    for (i=0; i<n; i++)
    {
        pos = i;
        for (j=i+1; j<n; j++)
        {
            if (bt[j] < bt[pos])
                pos = j;
        }
    }
```

Date :- 22/06/2021
Abhatt

```

temp = bt[i];
bt[i] = bt[pos];
bt[pos] = temp;
temp = p[i];
p[i] = p[pos];
p[pos] = temp;
}

wt[0] = 0;
for (i = 1; i < n; i++)
{
    wt[i] = 0;
    for (j = 0; j < i; j++)
        wt[i] += bt[j];
    total += wt[i];
}

avg_wt = (float)total/n;
total = 0;
printf("\nProcess\t Burst Time\t Waiting Time\t Turnaround Time");
for (i = 0; i < n; i++)
{
    tat[i] = bt[i] + wt[i];
    total += tat[i];
    printf("\np%d\t\t %d\t\t %d\t\t %d", p[i], bt[i], wt[i], tat[i]);
}

avg_tat = (float)total/n;
printf("\n\nAverage Waiting Time = %.f", avg_wt);
printf("\n\nAverage Turnaround Time = %.f\n", avg_tat);
}

```