NAME = Ayush Rawat

University Roll No = 1921095

Student Id = 19042075

Class Roll. No = 14

```
#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;
  }
  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;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("\nProcesst Burst Time twaiting Timet
        Turnaround Time");
for (i=0; i<n; i++)
{
ta t[i] = bt[i] + wt[i];
total += tat[i];
printf("%p%d\t%d\t%d\t%d", P[i], bt[i],
        wt[i], tat[i]);
}
avg_tat = (float) total/n;
total = 0;
```

```
avg_wt = (float) total / n;
total = 0;
printf("\n Process\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("\n%d\t%d\t%d\t%d", p[i], bt[i],
        wt[i], tat[i]);
}
avg_tat = (float) total / n;
printf("\n\n Average Waiting Time = %f", avg_wt);
printf("\n Average Turnaround Time = %f\n", avg_
        tat);
}
```

```
Enter number of process:4
nEnter Burst Time:np1:10
p2:9
p3:1
p4:4
nProcesst Burst Time tWaiting Time
tTurnaround Timenp3tt 1tt 0ttt1np4
tt 4tt 1ttt5np2tt 9tt 5ttt14np1tt
10tt 14ttt24nnAverage Waiting Time
=5.000000nAverage Turnaround Time=
11.000000n

...Program finished with exit code
```

NAME = Ayush Rawat
University
Roll No. 1921045
Student Id = 19042075
Class Roll No = 14

Q1 ANS

```
#include <assert.h>
#include <ctype.h>
#include <limits.h>
#include <math.h>
#include <stdbool.h>
#include <stddef.h>
#include<stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include<string.h>

char* readline();
char* ltrim(char*);
char* rtrim(char*);
char** split_string(char*);

int parse_int(char*);

int minimumAverage(int customers_rows, int customers_column, int** customers) {

}
```

```
int main()
{
    FILE* fptr = fopen(getenv("OUTPUT_PATH"), "w");

    int n = parse_int(ltrim(rtrim(readline())));

    int** customers = malloc(n * sizeof(int*));

    for (int i = 0; i < n; i++) {
        *(customers + i) = malloc(2 * (sizeof(int)));
    }

    char** customers_item_temp = split_string
        (rtrim(readline()));

    for (int j = 0; j < 2; j++) {
        int customers_item = parse_int(*
            (customers_item_temp + j));
        (*(customers + i))[j] = customers_item;
    }
}

int result = minimumAverage(n, 2, customers);

fprintf(fptr, "%d\n", result);

fclose(fptr);
}
```

```
return 0;
}

char * readline(){
    size_t alloc_length = 1024;
    size_t data_length = 0;
    char* data = malloc(alloc_length);
    while (true){
        char* cursor = data + data_length;
        char * line = fgets(cursor, alloc_length - data
                            _length, stdin);
        if (!line){
            break;
        }
        data_length += strlen(cursor);
        if (data_length < alloc_length -1 || data
            [data_length -1] == '\n'){
            break;
        }
        alloc_length <<= 1;
        data = realloc(data, alloc_length);
```

```
if (!data){
    data = "\0";
    break;
  }
}

if (data[data_length-1] == '\n'){
    data[data_length-1] = '\0';
    data = realloc(data, data_length);
    if (!data){
      data = "\0";
    }
  }else{
    data = realloc(data, data_length+1);
    if (!data){
      data = '\0';
    }else{
      data[data_length] = '\0';
    }
  }
  return data;
}

char *ltrim(char* str){
```

```
if (!str){
    return '\0';
}
if (!*str){
    return str;
}
while (*str != '\0' && isspace (*str)){
    str++;
}
return str;
}
char * strim (char* str){
if (!str){
    return '\0';
}
if (!*str){
    return str;
}
char * end = str + strlen (str)-1;
while (end > str && isspace (*end)){
    end--;
```

```
}

*(end+1) = '\0';

return str;

}

char** split_string(char* str){
    char** splits = NULL;
    char* token = strtok(str," ");
    int spaces = 0;
    while(token){
        splits = realloc(splits, sizeof(char*) * ++spaces);
    if(!splits){
        return splits;
    }
    splits[spaces-1] = token;
    token = strtok(NULL," ");
    }
    return splits;
}

int parse_int(char* str){
    char* endptr;
    int value = strtol(str, &endptr, 10);
```

```
if (endptr == str || *endptr != '\0') {
    exit(EXIT_FAILURE);
}

return value;
}
```

```
5
1 4
1 9
3
2 4
3


4
9
..Program finished with exit code
```