NAME - Ritik
Course - B.Sc (IT)
Semester - 2nd
Student Id - 20051095
Roll no. - 2023086
Paper name -
Paper Code -

```
#include <string.h>
#include <math.h>
#include <stdint.h>
#include <limits.h>
#include <stype.h>
#include <stdlib.h>
#include <assert.h>

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

int parse_int(char*);

int minimumAverage(int customers_rows,
int customers_columns, 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(n * sizeof(int*));

char** customers_item = item = 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);
fprint(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){
```

~~break; readline() {~~
while (true) {
char* ~~data~~ cursor = data + data_length;
char* line = fgets (cursor, alloc_length -
data_length, stdin);

if (!line) {
break;
}
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') {
dat [data_length -] = '\0';
data = realloc (data, data_length);

if (!data) {
data = '\0';
}
} else {
data = realloc (data, data_length+1);

```
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 * rtrim(char* str) {
        if (!str) {
            return '\0';
}
    if (!*str) {
        return '\0';
}
    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 (cudptr->str[0] * cudptr != '\0') {
        exit (EXIT_FAILURE);
    }
    return value;
}
```