Date.
Page No.

Name- Arman Tiwari
University roll no- 2023037
Student ID- 20051096
Father's Name- Sunil Tiwari
Course- Bsc IT(A)
Semester- 2
Paper name- OS Lab

Q1

```
#include<stdio.h>
#include<string.h>
#include<math.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("OUT_PATH"),
                "w");
int n = parse_int(ltrim(rtrim(readline())));
```

Arman

```
int** customers = malloc(n* sizeof(int*));
for (int i=0; i<n; i++){
    *(customers+i) = malloc(2* (sizeof(int)));
    char** customers_item_temp = split(rtrim
    (readline()));
    for (int j = 0; j<2; j++){
        int customers_item = parse_int(*(
        customers+i)+j) = customers_item;
    }
}
int result = minimumAverage(n, 2, customers);
fprintf(fptr, "%d\n", result);
fclose(fptr);
return 0;
}
char *ltrim(char* str){
    if(!str){
        return "\0";
    }
    if(!*str){
        return str;
    }
    while(*str!='\0' && isspace
return str;
}
char* rtrim(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;
  }
  split[spaces-1]=token;
  token=strtok(NULL," ");
}
return splits;
}
```

Date.
Page No.

```
int parse_int(char* str){
    char* endptr;
    int value = strtol(str, &endptr, 10);
if(endptr == str || *endptr != '\0'){
    exit(EXIT_FAILURE);
}
return value;
}
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* cursor = data + data_length, stdin);~~
    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[data_length] = '\0';
  }
}
return data;
}
```