Na,me-Prashant Rokaya

Course Bsc IT 4 sem

Roll no-1922043

```
#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<std.lin.h>
#include<string.h>
char* readline();
char* ltrim(char*);
char* rtrim(char*);
char** split_string(char*);

int parse_int(char*);

/*
* complete the 'minimumAverage' function below.
*
```

```
* The function is expected to return an INTEGER.
* The function accepts 2D_INTEGER_ARRAY customers as
parameter.
*/

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(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';
```