

NAME = Ayush Rawat
University

(JNPM) 101

Roll No. 1921045

3

(Student Id = 0919042075) roll no = 6191-33117

Class Roll No = 14

Q1 #include <assert.h> // solution = 86.00 marks

ANS # include <ctype.h>

include <limits.h> // (n > i + m) = 60

include <math.h> // (l + r) / 2 = (l + r) / 2

include <stdbool.h>

include <stddef.h> // solution = 100

include <stdint.h>

include <stdlib.h> // (l + r) / 2

include <stdlib.h> // (l + r) / 2

include <string.h>

char * readLine();

char * trim (char *);

char * trim (char *);

char ** splitString (char *);

int parseInt (char *);

int minimumAverage (int customers_200s, int o) {
 return (customers_200s * o) / customers_200s;

cout << "Customer Average = " << minimumAverage << endl;

int main()

3

FILE * fptr = fopen (getenv ("OUTPUT_PATH"), "w").

```
int n = parse_int(trim(trim(readline(c))));
```

`int *Customer = malloc (n * sizeof (int *))`

```
for (int i = 0; i < n; i++) {
```

* $(customers + i) = \text{malloc}(\{ \text{char}^{\text{long}} > 3 \text{ byte} \})$ $\{ \text{char}^{\text{long}} > 3 \text{ byte} \}$ $\{ \text{size of (int)} \}$

chart** customers-item-temp = Split-Stage

• `(trim (deadline)))`

```
int customers-item = parseint(*
```

(customers-item-temp[ij])) ;
{
 cout << "Total cost of item " << i << endl;

$$((\text{customers} + i)(j_{\text{last}})) = \text{customers.item}_{i+j}$$

3 $\left(\begin{smallmatrix} * & 0 \\ 0 & 1 \end{smallmatrix}\right)$ Basis \rightarrow $\{q_1, q_2\}$ \in $\mathbb{R}^{2 \times 2}$

$$y = \left(\frac{1}{\sqrt{2}} \sigma_0 + \frac{i}{\sqrt{2}} \sigma_1 \right) |0\rangle$$

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

```
fprintf(fp, "%d\n", result);
```

`f.close(fptr);`

return 0;

}

char * readline()

size_t alloc_length = 1024;

size_t data_length = 0;

char * data = malloc(alloc_length);

while (true){

(alloc_length, pfb) = fgets(data + data_length, pfb)

char * cursor = data + data_length;

char * line = fgets(cursor, alloc_length - data_length, stdin);

if (!line){

(alloc_length, pfb) = fgets(data + data_length, pfb)

data_length += strlen(cursor);

if (data_length < alloc_length - 1 || data

[data_length - 1] == '\n') {

data = realloc(data, alloc + length);

break;

alloc_length <= 1;

data = realloc(data, alloc + length);

```

if (!data) {
    data = "\0";
    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_length] != '\0') {
                data[data_length] = '\0';
            }
        }
    }
    return data;
}

```



REDMI NOTE 8 (char * ltrim (char * str)) {

A QUAD CAMERA

if (!str) {

 return '\0';

}

if (!*str) {

 return str;

}

if ((str == NULL) || (*str == '\0') && isspace(*str)) {

 while ((*str != '\0') && !isspace(*str))

 (str++) ;

}

 if (str == NULL) return '\0';

 return str; }

char * trim(char * str) {

 if (!str) {

 return '\0';

 int start = [1 - strlen(str)] str++;

 if (!(*str)) {

 return str;

 }

 char * end = str + strlen(str) - 1;

 while (end >= str && !isspace(*end)) {

 end--;



REDMI NOTE 8

AI QUAD CAMERA

16MP + 8MP + 2MP + 2MP

3

* (emd + 1) = '16'

return str;

3

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; }

3

splits [spaces - 1] = token;

token = strtok (NULL, " ");

3

return splits;

3

int parse_int (char * str) {

char * endptr;

int value;

value = strtol (str, &endptr, 10);



REDMI NOTE 8



AI QUAD CAMERA

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

}

```
return value;
```

}



REDMI NOTE 8
AI QUAD CAMERA

```
5
1 4
1 9
3
2 4
3
4
9
...Program finished with exit code
```