

Name - Arjan Singh Negi
Student id - 20051012

```
(1) #include <assert.h>
#include <ctype.h>
#include <limits.h>
#include <math.h>
#include <stdbool.h>
#include <stddef.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 minAvg(int customer_row, int customer_columns, int** customer)
```

```
int main()
```

```
{
    int main()
```

```
FILE * fptr = fopen("Output-path", "w")
```

```
int n = parse - int(ltrim(rtrim(readline())));
```

```
int** customers = malloc(n * size of(int*))
```

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



```
* (customer + i) = malloc (2 * size of (int));
```

```
char ** @ customer - item - temp = split - string  
(trim(readline));
```

```
for (int j = 0; j < 2; j++) {
```

```
int customer - item = parse - int (* (customer - item -  
temp + j))
```

```
((customer + i) + j) = customer [k];
```

```
}
```

```
}
```

```
int result = minimum Ave (n, 2, customer);
```

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

```
fclose (ptr);
```

```
return 0;
```

```
}
```

```
char * readline() {
```

```
size_t alloc length = 1024;
```

```
size_t data_length = 0;
```


Sunil
Page _____
Date _____

```
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 = "NO";
```

```
    }
```



```
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;
```

```
}
```


return data;

char* trim(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 str;

}

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

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

end--;

int parse = int (char *str) {

char *endptr

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

if (endptr == str || *endptr != '0')

{

exit (Exit failure);

}

return value;

}