Name - Shubham Gosai Section - B.

Father name - Mr. Chardal pal sing Type - practical

Course - BSc IT

SID - 20051114

Campus - Dehradun.

Ques 1.

Ans

```
#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>
#include<
char * readline ();
char * ltrim (char*);
char * rtrim (char*);
char ** split string (char*);

int parse_int (char*);
/*
* Complete the minimum Average function
*/
```

```
int minimumAverage(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(*(customer_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) {
  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 + 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) {
     retur '\n';
   }
   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;
}
  splits[spaces - 1] = token
  token = strtok(NULL, " ");
}
  return splits
}
int parse_int(char* str) { char *endptr
 int value = strtol(str, &endptr, 10);
if (endtr == str || *endptr != '\0') {
 exit(EXIT_FAILURE);
}
return value
}
```