NAME - ANISHA RAWAT

STUDENT ID - 20422001

UNIVERSITY ROLL NO. - 2023030

COURSE - BSC IT - 2B

Q.1. Problem Question :-

```
# 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 < stdlib. h>
# include < string. h>

char* readline ();
char * ltrim (char*);
char* rtrim (char*);
char ** split_sring(char*);

int ~~phase~~ parse_int( char *);
int minimumAverage(int customers_rows, int customers_columns,
                    int ** customers) {

}
```

Anisha
22/06/21

```
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);
  fprint(fptr,"%d\n", result);
  fclose(fptr);
  return 0;

}
```

Anisha
22/06/21

```
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;
   }
 }
```

Rnisha
22/06/21

```
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;
}
  char* ltrim(char * str) {
   if(!str){
   return str;
  }
  while (*str != '\0' && issapace(*str)) {
     str++;
  }
     return str;
  }
```

22/06/21

```
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;
}
```

Rushi
22/06/21

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

Anisha
22/06/21