Name - Nirmal Tafwal
Uni. Roll No - 2023073
Course - B.Sc IT
Sec - A

1)
```
# include < stdio. h >
# include < ctype. h >
# include < limits. h >
# include < math . h >
# include < stdbool . h >
# include < std def . h >
# include < stdin . h >
# include < stdlib. h >
# include <string . h >

char * readline ();
char * . l trim (char *);
char * r trim (char *);
int parse - int (char *);

int main ( )
{
File * f ptr = f.open (getenv ("Output _ Path"), "w");
int n = parse in (l trim (r trim(readline ())));
 int ** customers = malloc (n* size of (int*));
 for (int i = 0; i < n; i++)
{
 * (customers + i) = malloc (2 * (size of (int*)));
char ** customers - item - temp = split - string (r trim (
         readline)
 for (int J = 0 ; J < 2 ; J++)
 { int customers - item = parse - int (* (customers -
    item - temp ")));
```
[illegible]

```
((customers + i) + J) = customers - item; } }
  int result = minium Average (n, 2, customers );
    fprintf (fptr, "% d\n", result);
     fclose (fptr);
      return 0;
  }
    char * readline (){
   size - t alloc - length = 1024 ;
    size 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 ; }
     if (data length < alloc length - 1 // dat [data
          length - 1) = '\n'8)
       { break ; }
     alloc _ length << = 1 ;
     data = realloc (data, alloc _ length);
       if (! data)
       data = '\0' ;
     break ;
       }
    } if (data (data length - 1] == '\n'){
      data [data - length - 1] = '10' ; }}
```

```
else
{
 data = realloc(data, data length +1);
 if(!data)
{
  data= "\0';
}
 else
{
  data [data_length] = ~~0~~'\0';
 }
}
 return data;
 }
 char* trim (char* str)
 { if(! str)
   { return '\0';
   }
   if (*str)
   { return str;
   }
  while (*str != '\0' && isspace (*str))
   { str++;
   }
   return str;

  }
  char* strim (char *str)
  { if(!str)
  { return '\0';
  } if (! *str)
  { return str;
  }
  char * ind = str + strlen (str)-1;
```

```
end --) }
*(end+1) = '\0';
return 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 (endptr == str || *endptr != '\0') {
   exit (EXIT_FAILURE);
}

   return value;
}
```