Name - Lovejeet singh mehta
Std. ID - 20051123
Course - B.Sc IT
Subject - PBI 202 (Operating system)
Ans ①

```
#include<bits/stdc++.h>

using namespace std;

string ltrim(const string&);
string rtrim(const string&);
vector< string>split(const string&);

/*
* Complete the 'minimumAvarge' function below.
*
* The function is expected to return an INTEGER.
* The function accepts 2D-INTEGER-ARRAY
    coustomer as parameter.
*/

 ~~int minimumAvarge(getenv("OUTPUT~~
 int minimum Avrge (vector<vector<Int>> ~~cou~~
   costomers){
 }

  int main( )
{

   ofstream fout(getenv("OUTPUT_PATH"));

   string n_temp;
   getline (cin,n_temp)));
```

DATE

```
vector< vector<int>> customers(n);

for(int i=0;<n;i++) {
    customers[i]~~[j]~~.resize(2);

  string customers_raw_temp_temp);
  getline (cin,customers_row_tem_temp
   vector < string> customers_raw_temp=
  split (rtrim (customers_raw_temp_temp

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

  int customers_row_item=stoi(customers_
           raw_temp[j]);

  customers[i][j] = customers_row_item;
  }

}

  int result = minimum Avarage(customers);
 fout <<result<< "\n";
  fout.close();

  return 0;
}
 string ltrim (const string& str){

  string s(str);

  s.erase()
```

```
S. begin(),
 find_if(s.begin(), s.end(), not1(ptr
   _fun<int, int>(isspace)))
 );
   return s;
 }

 string rtrim(const string &str){
  String s(str);
 S.erase (
   find_if(s.rbegin(), s.rend(), not
    1(ptr_fun<int, int>(isspace))).base(),
    s.end()
  );
   return s;
 }

vector <string> split(const string&s
{
  vector<string> tokens;
```

DATE

```
String :: size _type start =0;
string :: size - type end =0;
while ((end =str. find (" ",start))!
  = string::npos ) {
tokens .push_back (str. substr(start,
  end - start ));
 start =end +1;
}
 tokens.push - back (str. substr(start));
return tokens;
}
```