|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | | | |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | | **Course:** | **ITC Lab** | **Course Code:** | **CS-101** | |
| **Program:** | **BS (Computer Science)** | **Semester:** | **Fall2017** | |
| **Duration:** | **150 Minutes** | **Total Marks:** | **40** | |
| **Paper Date:** | **05-Dec-2017** | **Weight** | **40%** | |
| **Section:** | **D** | **Page(s):** | **2** | |
| **Exam:** | **Final Term** | **Roll. No.** |  | |
|  | |  | | | | |

**Instructions:**

1. Understanding the question paper is also part of the exam, so do not ask any clarification.
2. Make sure to switch off your mobile phones before the Exam starts.
3. No USB’s are allowed. No INTERNET is allowed. Please see that the area in your threshold is clean. You will be charged for any material which can be classified as ‘helping in the paper’ found near you.
4. Talking/Discussion is not allowed. It is your responsibility to protect your code and save it from being copied. If you don’t protect it all matching codes are considered copy/cheating cases.
5. Write a C++ Program to input an English Sentence and output the **count of two and three letter words**. **(10 marks)**

**Input:**

This is ITC Final Exam for section D

**Output:**

Count of two letter words =1

Count of three letter words = 2

1. Write a C++ program to sort a **2D array in ascending order using selection sort** logic.  **(15 marks)**

To perform this task, you will write a function Minimum to find the index of the minimum number in a 2D array starting from a particular cell of the 2D array. If there are multiple entries, then return the index of the first smallest entry.

**void Minimum(int Arr[][COLS], int StartX, int StartY, int &MinX, int &MinY)**

Now write a Funtion **SORT2D** that will get a 2D array as a parameter and will sort it using above Minimum Function. Carefully think about the prototype of the function SORT2D.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 56 | 10 | 13 | 20 | 99 |
| 21 | 9 | 22 | 19 | 12 |
| 3 | 8 | 11 | 18 | 23 |
| 24 | 7 | 15 | 17 | 4 |
| 5 | 6 | 14 | 25 | 16 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 |
| 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 56 | 99 |

**2D Array before calling SORT2D 2D Array after calling SORT2D**

1. Write a C++ program to input **N from user and print** the following pattern. **(5 marks)**

If N=5 then your output would be:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 10 | 11 | 20 | 21 |
| 2 | 9 | 12 | 19 | 22 |
| 3 | 8 | 13 | 18 | 23 |
| 4 | 7 | 14 | 17 | 24 |
| 5 | 6 | 15 | 16 | 25 |