| | |
|---|---|
| **Started on** | Friday, 10 January 2025, 2:05 PM |
| **State** | Finished |
| **Completed on** | Friday, 10 January 2025, 4:46 PM |
| **Time taken** | 2 hours 40 mins |
| **Grade** | **100.00** out of 100.00 |

**Question 1**

Complete

Mark 10.00 out of 10.00

Create a file with the name **ROLLNO_week1_q1.cpp** and write a C++ program to print numbers with the following requirement.

- Ask user the start number
- Print five numbers
- Ask the user again for the end number
- Print numbers until the end number
- You may not use any pointers or reference in your program
- You may not use any global variables
- All incrementing of the number should be strictly done inside the `get_num_after_increment` and not anywhere else.

Use the following template:

```
#include <iostream>

using namespace std;

int get_num_after_increment(int k)
{
// You must increment only in this function.
// Do not change the signature of this function. Do not use a global variable.
}
int main()
{
// Complete this code
return 0;
}
```

The following is a sample run:

```
Enter a start: 12
12
13
14
15
16
Printed five numbers. Enter end: 26
17
18
19
20
21
22
23
24
25
26
```

23CS30048_week1_q1.cpp

Comment:

**Question 2**

Complete

Mark 90.00 out of 90.00

The following link contains a data dump of users that got leaked from a server.

https://moodlecse.iitkgp.ac.in/moodle/pluginfile.php/19621/question/questiontext/1152/2/116/users.txt

Unfortunately, the data got shuffled due to a bug in the software creating the dump. Your task is to fix the overall data dump and analyse it.

The following is some information related to the data.

- Each line in the file corresponds to one user. The user information is comma-separated.
- The data contained in each line is the username (prefixed with user_), company the person works for, password, salary for each month (Eg. Jun-8969 means the salary for the month June is 8969).
- Salaries of the months got shuffled during the data dump.
- For some users, salary details of some months are missing which is expected as people may join a company any month in a year and might be on leave for few months.

Create a file with the name **ROLLNO_week1_q2.cpp** and write a C++ program that.

- Reads the given file (users.txt) and parses each user.
- Remove the 'user_' prefix from the username.
- The salary details of users got corrupted during dumping. Round off all salaries to the nearest integer and sort by month. Compute the mean salary for each user and add this information to the end of each line. Example: Mean_Salary-2000.
- Suffix a weak password (length less than 9) with __WEAK_PASSWORD.

Once the data is properly cleaned, write back to a new file called *new_users.txt* and **upload** it as part of the solution.

**Example:**

If the corrupted file contains the following line that corresponds to an user.

user_burpthealphabet,intel corporation,buri78, Feb-10.1,Jan-20

The line after cleaning and adding extra information will look like the following:

burpthealphabet,intel corporation,buri78__WEAK_PASSWORD,Jan-20,Feb-10,Mean_Salary-15

**Additionally,** analyse the data and print the the following information about the data dump on executing your program.

- The average length of the password (without the __WEAK_PASSWORD suffix you might have added later).
- Print the longest and shortest username (not counting the length of user_ prefix).
- Print the top-5 most frequent companies the users belong to and the frequencies.
- What is the minimum salary someone might have received? What is the maximum salary someone might have received?
- Which company pays the highest average salary?
- Which company pays the lowest average salary?
- Which month of the year most users have received a salary/worked?

---

# Helper code to read a file

```
#include <iostream>
#include <string>
#include <fstream>

using namespace std;

int main(int argc, char *argv[])
{
    string line;

    ifstream infile("users.txt");

    while (getline(infile, line))
    {
       cout << line << endl;
    }
    infile.close();
    return 0;
}
```

## Helper code to write to a file

```
#include <iostream>
#include <string>
#include <fstream>

using namespace std;

int main(int argc, char *argv[])
{
    ofstream outfile;
    outfile.open("new_users.txt");
    outfile << "Hello";
    outfile.close();
    return 0;
}
```

- 23CS30048_week1_q2.cpp
- users.txt
- new_users.txt

Comment:

◄ Week-0: C Programmig Practice

Jump to...

Unit Testing ►