MATEMATISKA INSTITUTIONEN
STOCKHOLMS UNIVERSITET
Avd. Beräkningsmatematik
Kursledare: Lars Arvestad
Examinator: Lars Arvestad

Tentamen i
DA4007 Programming techniques II
7.5 hp
2024-10-23

- Part A consists multiple-choice questions, worth 1 point per question, where at least one answer option is correct. If you answer incorrectly or do not select the exact number of correct options, you will receive zero points for that question.
- Part B consists of questions with varying point values.
- Grade boundaries: E: 10, D: 12, C: 14, B: 16, A: 18, out of a maximum of 20.

Part A: multiple choice

Please gather the answers for Part A on a single answer sheet.

1. Why would you use *git*?

A. To manage different versions of a project's source code.
B. To compile C++ code to an executable file.
C. To run Unix commands.
D. To visualize source code dependencies.

2. One of the principles in Unix is "Everything is a text-file". Which justification is there for this principle?

A. Unix is old, and that is how data was stored in the early days of computing.
B. It is a good way to save disk-space.
C. Transferring data over the internet becomes easy then.
D. It makes it easy to do computations using regular Unix tools.

3. What does "piping" mean in Unix?

A. Output is saved in a named file.
B. Output from one program is moved to another program, without saving in a file.
C. Input is taken from a different directory in the file system.
D. An input file is empty, so a program reads from the keyboard instead.

4. What is the C++ mechanism for managing run-time problems and errors called?

A. Segmentation fault
B. Return values
C. Recursion
D. Exceptions

5. What is a "header file"?

A. A file with text explanations of data, separate from the data itself.
B. A file describing how a C++ project is compiled.
C. A file containing C++ declarations.
D. A file that is executable.

Part B: general questions

Please use a separate sheet of paper for each question in Part B.

6. You are given a large data file `starfield.csv` containing lines of numbers, with each line representing an observation. Write Unix commands to solve the following questions.

A. Show how to compute the number of lines in the data file. (1p)

B. You are told that some observations may have been reported several times, giving some lines to appear more than once. Show how to count the number of duplicated lines there are in the file. (1p)

C. Show how to determine whether the number "1.2345" appears in the 100 first lines of the data file.(1p)

7. Write the C++ program `rev.cpp` that prints the first command-line argument backwards. (3p)

After compiling to the executable `rev`, it should work like this:

```
$ ./rev DA4007
7004AD
$ ./rev Stockholm University
mlohkcotS
$ ./rev
Usage: rev <string>
```

8. Write a class `Point` in C++ for storing points in the plane. Each object should have floating point attributes for x and y. There should be a method named `magnitude` that returns $\sqrt{x^2 + y^2}$. Ensure that the following code would work with your class. (3p)

```
int main() {
  Point p(2.0, 3.0);
  cout << p.x << ", " << p.y << endl;
  cout << "Magnitude: " << p.magnitude() << endl;
}
```

9. The C++ function `n_uppercase`, below, computes the number of uppercase letters in a string. However, there are two mistakes in the code: describe them. (2p)

```
n_uppercase(string &s) {
  int n=0;
  for (int i=1; i<s.size(); i++) {
    if (s[i] >= 'A' && s[i] <= 'Z') {
      n++;
    }
  }
  return n;
}
```

10. In this problem we want a program that can sum lines with floating point numbers given in a text file.

A. Write the program `sum.cpp` that reads floating point numbers from stdin and prints their sum. (2p)
After compiling to the executable `sum`, it should work like this:

```
$ ./sum < empty_file.txt    # File size is zero
0
$ ./sum < data.txt
6.3
```

In the second example, `data.txt` contains three lines with the numbers 1.0, 2.1, and 3.2.

Hints: You might want to use `getline(inputstream, destinationstring)` for reading the data and `stof(s)` for converting a string `s` to a float. Assume the input file has a single float per line.

B. Handle the error `invalid_argument` that occurs when `stof` cannot convert a string to a number. Instead of crashing, the program should exit with the message “Bad input” written to stderr. (2p)

Example:

```
$ ./sum < sum.cpp
Bad input
```