

Key

You may consult one 8.5x11 inch sheet containing any information about the material covered in this course. This sheet must be turned in attached to the exam. No other materials, written or otherwise, are allowed other than the copy of the exam you are using and a pen or pencil. All questions about the exam should be addressed to the instructor or a teaching assistant. All questions in the exam should be answered on the exam or on the lined paper provided. Use only a blue or black pen or a dark (no. 2) pencil. (Note: The exam is divided into 175 points.)

1) Write the UNIX commands to do the following:*(10 Points, 2 each)*a) Copy the file "niacin.cc" from your home directory to your class directory. `cp niacin.cc cs1501`b) List the files in your current directory and show the size of each. `ls -l`c) Change the name of file "pellagra.cpp" to "pellagra.cc". `mv pellagra.cpp pellagra.cc`d) Find out what directory you are in. `pwd`e) Compile the file "dermatitis.cc" using the Gnu G++ compiler. `g++ dermatitis.cc`**2) Write the output of the following code fragments.***(30 points, 5 each)*

a)

```
int j = 4;
while (j > 0)
{
    cout << j << ' ';
    j--;
}
```

`4 3 2 1`

b)

```
for (int z=0; z<5; z+=2)
{
    cout << z << ' ';
}
```

`0 2 4`

c)

```
int w=0;
while (w < 10)
{
    w += 2;
    if (w%3 == 0)
        cout << "left";
    else
        cout << "right";
}
```

`rightrightlefttrightright`

```
d) int s = 20,
    sum = 0;
    for (; s>0; s-=3)
    {
        if (s % 2 == 0)
            sum = sum + s;
    }
    cout << sum;
```

44

```
e) int out = 3,
    in;
    while (out > 1)
    {
        in = 2;
        while (in > 0)
        {
            cout << out << in << endl;
            in--;
        }
        out--;
    }
```

32
31
22
21

```
f) for (int wrap=8; wrap<12; wrap += 2)
    {
        for (int wrapped=1; wrapped<3; wrapped++)
        {
            cout << wrap << wrapped << endl;
        }
    }
```

81
82
101
102

3) Given the contents of the input file, what is the output of the following code?
(5 points)

```
ifstream numsIn;
double num,
    sum=0;
numsIn.open("stuff.dat");
numsIn >> num;
while (numsIn)
{
    if (num > 5)
        sum += num;
    numsIn >> num;
}
cout << sum;
```

"stuff.dat"1
7
5
2
8
9
-4
3

24

4) What is the output of the following program? (16 points)

```

#include <iostream>
#include <string>
using namespace std;

const string FOUR = "dixi,";
string ChangeUp(string, string &, string);

int main()
{
    string one    = "veni,",
          two     = "vidi,",
          three   = "vici,",
          four;

    cout << one << two << three << FOUR << endl;
    four = ChangeUp(one, two, three);
    cout << one << two << three << four;

    return 0;
}

string ChangeUp(string a, string & b, string c)
{
    a = b;
    b = c;
    c = FOUR;
    return a;
}

```

```

veni,vidi,vici,dixi,
veni,vici,vici,vidi,

```

5) On a sheet of lined paper, write code fragment to do the following: (10 points)

Sum the numbers (and just the numbers) in the file “nonsense.dat”. The file contains a word (with no spaces), followed by a space, followed by a number, followed by a space, etc. For example:

spam 10 sew 13 up 3 down 5 ... rime 3 end-of-file

```

ifstream inFile;
string s;
int num, sum=0;

inFile.open("nonsense.dat");
if (!inFile) // optional on test
{ // optional on test
    cout << "no file"; // optional on test
    exit(1); // optional on test
} // optional on test
inFile >> s >> num;
while (inFile)
{
    sum += num;
    inFile >> s >> num;
}

```

6) The function `Lefty` shifts integers 100 places to the left on the number line (i.e. subtracts 100 from them) and returns the resulting integer. The prototype of `Lefty` is as follows:

```
int Lefty(int num);
```

Write a call to `Lefty` so that the following variable has its value reduced by 100.

(5 points)

```
int theNum = 987;
```

```
theNum = Lefty(theNum);
```

7) On a sheet of lined paper, write the header and body of the function `Lefty` as described in #6.

(10 points)

```
int Lefty(int num)
{
    return num - 100;
}
```

8) On a sheet of lined paper, write a non-void function that receives 3 numeric grades, drops the lowest grade and returns the average of the remaining two grades. (Note: It is not necessary to write a function prototype, #includes or any other code except the code that makes up the function.)

(35 points)

```
int Average(int grade1, int grade2, int grade3)
{
    if (grade1<=grade2 && grade1<=grade3)
        return (grade2+grade3)/2;
    else if (grade2<=grade1 && grade2<=grade3)
        return (grade1+grade3)/2;
    else
        return (grade1+grade2)/2;
}
```

Example calls:

```
cout << Average(100,200,300) << endl;
cout << Average(200,300,100) << endl;
cout << Average(300,100,200) << endl;
cout << Average(100,100,200) << endl;
cout << Average(100,200,100) << endl;
cout << Average(200,100,100) << endl;
cout << Average(200,200,100) << endl;
cout << Average(100,200,200) << endl;
cout << Average(200,100,200) << endl;
cout << Average(100,100,100) << endl;
```

Output:

```
250
250
250
150
150
150
200
200
200
100
```

9) On a sheet of lined paper, write a complete and correct C++ program that reads and process all data in a file of unknown length called “grades.dat”. The file consists of information about grades for students in a class. This information consists of student names and 3 numeric grades per student name. The file has the following organization. (Note: The grades of a student follow a students name and that each student always has 3 grades.)

```
student-name
grade grade grade
student-name
grade grade grade
student-name
grade grade grade
:
student-name
grade grade grade
end-of-file
```

The program must output to the monitor screen the name and average of each student. The average of a student is based on the two highest grades, the lowest being dropped. Your code should not calculate averages directly. Instead, the function from #8 should be called for each student and its return accepted as the average. (Note: Be sure and put a function prototype in your program, but it is not necessary to copy the function from #8 onto this program. It will be assumed to be present.)

(35 points)

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

using namespace std;

int Average(int grade1, int grade2, int grade3);

int main()
{
    string student;
    int grade1, grade2, grade3;
    ifstream inFile;
    inFile.open("grades.dat");
    if (!inFile) // optional on test
    { // optional on test
        cout << "no file"; // optional on test
        exit(1); // optional on test
    } // optional on test

    getline(inFile, student);
    while (inFile)
    {
        inFile >> grade1 >> grade2 >> grade3;
        cout << student << " has an average of "
             << Average(grade1, grade2, grade3)
             << endl;
        getline(inFile, student);
        getline(inFile, student);
    }

    return 0;
}

// ***** This copy of this function is ***
// ***** not necessary, having been ***
// ***** written in #8. It is just listed ***
// ***** here for completeness ***

int Average(int grade1, int grade2, int grade3)
{
    if (grade1<=grade2 && grade1<=grade3)
        return (grade2+grade3)/2;
    else if (grade2<=grade1 && grade2<=grade3)
        return (grade1+grade3)/2;
    else
        return (grade1+grade2)/2;
}
```