DAIICT – Mid-Semester Examination – Autumn 2020
IT603 – Programming

MM: 40 Time: 90 mins

Instructions:

1. There are 6 questions in this paper.
2. Marks for each question are mentioned against it
3. The question ID (Q1, Q2 …) must be mentioned in a separate line before beginning of its answer
4. Do NOT focus on getting the input / output right unless mentioned explicitly in the question.
5. To make it simpler, some questions have hardcoded data to work on. But your program logic should not assume hardcoded data and should work on data input in similar format
6. Using modern C++ features carries marks

Q1. See the program below and find out data type and qualifier you think is most appropriate for the variables. Take hints from comments. When run, the program gives following output:

4

```
50000 with interest of 8.3 compounded quarterly, will become 96469.73 after 8
years.
```

```
void Q1()
{
	............................... r{8.3}; // interest in percent, fraction, constant
value
	............................... p{}; // principal, large, only +ve amount, fraction
	............................... t{}; // duration in complete years, no fraction
	............................... n{4}; // compounded quarterly, constant value

	// example input:
	p = 50'000.00;
	t = 8;

	// a => future value
	auto a = p * pow((1 + (r/100.0f)/n), n*t);

	cout << p << " with interest of " << r << " compounded quarterly, will become
" << fixed << setprecision(2) << a <<" after " << t << " years.";
}
```

Max Word Limit: 700

Q2. Write a program to count the number of words in a sentence. You do not need to take input from user. Use the following and print the result to console. The program should not count extra spaces eg. around word ‘one’. For simplicity, assume that you do not need to deal with sentences with leading/trailing spaces.

8

```
char s[] = “The multi-word sentence could have more than  one  spaces between
words”;
```

Max Word Limit: 700

Q3. For a 2-d matrix, the right most column should have sum of rows and bottom row should have sum of columns. eg:

20	-10	20	30
20	20	20	60
-10	-20	20	-10
30	-10	60	80

Write a program to do so. Use following hardcoded matrix ‘m’ as input. Use another matrix for preparing the result as above. You do not need to print anything to output.

6

```
int m[3][3] {{20,-10,20},
             {20,20,20},
              {-10,-20,20}};
```

Max Word Limit: 700

Q4. Write a program to create a 2-d array of 4 rows x 50 columns and copy following strings to it in order of their length, first being the string with least length. Your program should work correctly regardless of order of the strings in array below.

8

```
char input[][50] ={"Little bigger than medium",
      "Small",
      "This is longest string of all",
      "Medium string"};
```

Max Word Limit: 700

Q5. Write a program to copy strings from following array to another 2-d array (of same size), but the strings should be copied as right aligned in target array (ie all spaces, if any, of a row should be in front and string’s null character should be in the last cell of its row).

6

```
char input[][50] ={"Little bigger than medium",
      "Small",
      "This is longest string of all",
      "Medium string"};
```

Max Word Limit: 700

Q6. In a word editor following user preferences can be set in a settings dialog by a user:
[] Auto Save
format:
[] docx

8

[] docm
[] doc
[] Auto Recover
[] Save to Local Disk
[] Save to Cloud

The format options are mutually exclusive, ie only one at a time will be selected and only if auto-save is selected. By default, the option will be docx.

Write program to save user’s choices from settings dialog using bitwise operators to a variable of smallest size possible. Also, the program should be able to read which bits are set from a variable to simulate showing user’s choices properly if user opens the settings dialog again.

Use following 1-d array as input coming to your code as which settings are set by user

int op[5] = {1,3,0,1,1} // from Left to Right -> 1= Auto Save, 3=> doc, 0 => Auto Recover, 1=> Save to Disk, 1=> Save to cloud.

Use this to set the bits of the variable you’ll save as settings (when user saves the settings). And then read from the same variable to set the above array (simulating showing of settings when dialog opens).

Max Word Limit: 700