LXU

(University of Choice)

MASINDE MULIRO UNIVERSITY OF SCIENCE AND TECHNOLOGY (MMUST)

MAIN CAMPUS

UNIVERSITY MAIN EXAMINATIONS 2021/2022 ACADEMIC YEAR

THIRD YEAR 2ND SEMESTER EXAMINATIONS

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

COURSE CODE: BCS 362
COURSE TITLE: GENERIC PROGRAMMING WITH C++

DATE: FRIDAY 22ND APRIL, 2022 **TIME:** 3:00 - 5:00 PM

INSTRUCTIONS TO CANDIDATES

THIS IS AN OPEN BOOK EXAMINATION
Answer Question **ONE (1)** and Any **OTHER 2** questions
Ensure your answers/ideas are clearly expressed
All your answers must be clearly numbered
Write in ink. Rough work can be done (in answer booklet) in pencil and will not be marked. Cross out any rough work.
Calculators, phones, tablets, computers not allowed

TIME: 2 Hours

MMUST observes ZERO tolerance to examination cheating

QUESTION ONE: COMPULSORY QUESTION [20 MARKS]

A **Huduma Centre** ticketing system issues service tickets to clients who visit the the centre. The ticket has a number and name of service sought. There are three tellers each dealing with a different service. The services are

1. NTSA
2. Registrar of Persons
3. Political Parties Things

The system maintains three queues, one for each service. The client chooses a service among the three services by entering a character, **N** for **NTSA**, **R** for **Registrar of Persons** and **P** for **Political Parties Things**. Elderly, sick, pregnant or disabled clients are given priority on the respective queue. After choosing a service, another prompt appears asking the user to choose priority, where user enters **1** to mean user should be prioritised or **0** for non-prioritised users. A prioritised user is placed in-front of the queue and a non-prioritised user is placed at the back of the queue. Write definition of a class template **HudumaQueue** that simulates the queues in the **Huduma Centre**. Include functions;

(a) **atFront()** that return ticket number at the front of the queue. [2 Marks]

(b) **join(...)** that adds a ticket number at the back or front of the queue, depending on the priority client is given. Declare this function inside the class template and provide its implementation outside the class template [3 Marks]

(c) **atBack()** that return ticket number for client at the back of the queue. [3 Marks]

(d) **leave()** that deletes a ticket number from the queue after a client has been served. [2 Marks]

(e) Write definition of the **main** function. In it

(1) Create the three queues. [2 Marks]

(2) Assume current NTSA ticket number is 1340, Registrar of Persons ticket number is 1234 and Political Parties Things ticket number is 2389. write a code fragment that will add a client to respective queues, such that your output resembles sample output below.

Note: *The screen clears after each ticket has been issued*

(i) Prompt user to choose the queue to join and capture user response. [2 Marks]

(ii) Prompt user to choose the queue to join and capture user response. [2 Marks]

(iii) Prompt user for priority status and capture user response. [2 Mark]

(iv) Add the user to the respective queue, and display ticket number. [2 Marks]

Ticketing System Sample Output 1. User input underlined

```
Enter
 N. NTSA
 R. Registrar of Persons
 P. Political Parties Things
Response: N
Priority Client? Enter
1. Priority
0. None Priority
Response: 0
Ticket Number 1341 NTSA Queue
Next Customer? If yes, type Y or y
Next: Y
```

Ticketing System Sample Output 2. User input underlined

```
Enter
 N. NTSA
 R. Registrar of Persons
 P. Political Parties Things
Response: R
Priority Client? Enter
1. Priority
0. None Priority
Response: 1
Ticket Number 1235 Registrar of Persons Queue
Next Customer? If yes, type Y or y
Next: -
```

Ticketing System Sample Output 3. User input underlined

```
Goodbye, see you again
```

QUESTION TWO [20 MARKS]

(a) Write a C++ function that estimates and returns the value of π using the formula

$$\pi = 2 \times \frac{2}{\sqrt{2}} \times \frac{2}{\sqrt{2+\sqrt{2}}} \times \frac{2}{\sqrt{2+\sqrt{2+\sqrt{2}}}} \cdots$$

This formula has infinite number of terms with increasing complexity, so you must multiply additional terms until the size of the next term is 1. [4 Marks]

(b) A positive integer is called a perfect number if it is equal to the sum of all of its positive divisors, excluding itself. For example, **6** is the first perfect number because $\mathbf{6 = 3 + 2 + 1}$. The next is $\mathbf{28 = 14 + 7 + 4 + 2 + 1}$. There are four perfect numbers less than **10,000**. Write a C++ program to find and display all these four numbers. [4 Marks]

(c) Write a program that simulates the working of a three wheel casino machine with numbers 1...10. The program should work as follows:

(i) The program randomly generates three integers for **wheel_1**, **wheel_2** and **wheel_3** - with each wheel taking a value in the range 1 – 10, 10 inclusive

(ii) The integers represent the numbers shown on the three wheels of a casino slot machine

The program outputs the payout for the three numbers as follows:

- If the three wheels display the same number, the payout is 80
- If exactly two wheels display the same number, the payout is 3
- If the three wheels display consecutive numbers (not necessarily in order), payout is 16
- The payout is 0 for all other combinations.

Examples of program execution are given below. Your output should match these examples as much as possible. [12 Marks]

Program execution : Example 1

```
---Three wheel values---
Wheel 1 -> 8
Wheel 2 -> 8
Wheel 3 -> 7
------------------------
Payout is 3
------------------------
```

Program execution : Example 2

```
---Three wheel values---
Wheel 1 -> 6
Wheel 2 -> 8
Wheel 3 -> 7
------------------------
Payout is 16
------------------------
```

QUESTION THREE [20 MARKS]

Given a year as a four digit integer, index of the day on which first of January for that year falls is given by

$$first = R\left(5(R((year-1),4)) + 4(R((year-1),100)) + 6(R((year-1),400)),7\right)$$

where $R(x,y)$ is a **mod** function that returns **x%y**. February has 29 days in a leap year. A leap year is a year that is divisible by 4. Given first lines of a program as

```
1 #include<iostream>
2 #include<string>
3 using namespace std;
4 string days[7] = {"Mon", "Tue", "Wed","Thu","Fri","Sat","Sun"};
5 string names[12] = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
    "Aug", "Sep", "Oct", "Nov", "Dec"};
6 int mons[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
7 int bday, bmonth, year;
```

(i) Write a function that returns the name of the day on which first January falls for a year stored in variable **year** [4 Marks]

(ii) Write a function that returns index of the day on which a birth day falls(birth day is stored in variable **bday**). [4 Marks]

(iii) Write definition of a function that displays the calendars of the birth month, the month before and the month after for the year given in variable **year**. [12 Marks]

Sample Output of calendar function. User input underlined

```
Enter day you were born:12
Enter your birth month:4
Enter year you want to check:2021

Calendar for March 2021
Mo  Tu  We  Th  Fr  Sa  Su
1   2   3   4   5   6   7
8   9   10  11  12  13  14
15  16  17  18  19  20  21
22  23  24  25  26  27  28
29  30  31  1   2   3   4

Calendar for April 2021
Mo  Tu  We  Th  Fr  Sa  Su
29  30  31  1   2   3   4
5   6   7   8   9   10  11
12  13  14  15  16  17  18
19  20  21  22  23  24  25
26  27  28  29  30  1   2

Calendar for May 2021
Mo  Tu  We  Th  Fr  Sa  Su
26  27  28  29  30  1   2
3   4   5   6   7   8   9
10  11  12  13  14  15  16
17  18  19  20  21  22  23
24  25  26  27  28  29  30
31  1   2   3   4   5   6
```

QUESTION FOUR [20 MARKS]

(a) An array can be declared and initialized as shown in Line 1 below.

```
1 array<double, 5> arr {6, 8, 2, 1, 3};
2 //call softmax(... , ...) function
3 for(int i : arr){
4   cout << i << " ";
5 }
```

(i) Write a generic function (**void softmax(... , ...)**) that performs softmax normalization of this array in place. Softmax normalization is given by

$$n_i = \frac{e^{x^i}}{\sum e^{x^i}}$$

where x_i is the value at index i in the array and n_i is the normalized value at index i. [8 Marks]

(ii) What would be the output given by the for loop, assume that function **softmax(... , ...)** is invoked at line 2. [4 Marks]

(b) Taylor series for **sin(x)** and **cos(x)** are defined as

$$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \cdots$$

$$cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \cdots$$

where x is the angle in radians and n is the number of repetitions. Given an angle in degrees (**deg**), we can convert it into radians (**rad**) using the formula

$$rad = \frac{deg}{180} \times \pi$$

where π is a constant whose value is approximately 3.14159.

(i) Write a generic function that receives angle in degrees in any numeric data type and return the angle in radians [1 Marks]

(ii) Using a **do...while** loop, write a generic function that receives angle in degrees in any numeric data type, and return the **sine** of that angle, when the error between the estimated and actual **sine** is below 10^{-3}. The actual **sine** can be found using the math library function **sin(x)** where **x** is degree in radians [4 Marks]

(iii) Using a **while** loop, write a generic function that receives angle in degrees in any numeric data type, and return the **cosine** of that angle, after 10 repetitions. [3 Marks]