CS101 Spring 2023 - Theory Quiz 2 - 27 March 2024
5 Questions, 25 Marks, (Instructor: Prof. Shivaram)

Roll Number	**SAMPLE**
Name	
Group	

QNo.	Marks	Graded by	Verified By	Student's Cribs
1 (4)				
2 (5)				
3 (5)				
4 (6)				
5 (5)				
Total				

Please read the following instructions carefully before you start.

- Write your roll number, name, and group number in the space provided. A paper without a roll number and name will NOT be graded.
- Write your answers neatly with a blue/black pen on this question paper itself in the space provided for each question. At the end, you must submit this paper to the invigilator.
- Rough pages will NOT be provided. Use the empty space in the margins.
- Please note that your answers should NOT include any programming concept that hasn't been covered in the class so far. If such answers are found, they shall NOT be graded.
- No clarifications will be provided on any questions. When in doubt, make suitable assumptions, state them clearly, and proceed to solve the problem. If your answer depends on any assumption you have made, the assumption must be explicitly stated in your paper.
- In some questions, you are provided code snippets. Assume that the code snippet is enclosed suitably within the main, correct header files are included, etc., and therefore, the code compiles.
- All the best!

Translation

शुरू करने से पहले कृपया निम्नलिखित निर्देशों को ध्यान से पढ़ें।

- दिए गए स्थान पर अपना रोल नंबर, नाम और ग्रुप नंबर लिखें। बिना रोल नंबर और नाम के पेपर को grade नहीं दिया जाएगा।
- इस प्रश्नपत्र पर ही प्रत्येक प्रश्न के लिए दिए गए स्थान पर अपने उत्तर नीले/काले पेन से साफ-सुथरा लिखें। अंत में आपको यह पेपर निरीक्षक के पास जमा करना होगा।
- रफ पेज उपलब्ध नहीं कराए जाएंगे. margin में दिए खाली जगह का उपयोग करें.
- कृपया ध्यान दें कि आपके उत्तरों में कोई भी प्रोग्रामिंग अवधारणा शामिल नहीं होनी चाहिए जिसे अब तक कक्षा में शामिल नहीं किया गया है। यदि ऐसे उत्तर पाए जाते हैं, तो उन्हें grade नहीं किया जाएगा।
- किसी भी प्रश्न पर कोई स्पष्टीकरण नहीं दिया जाएगा. जब संदेह हो, तो उपयुक्त धारणाएँ बनाएं, उन्हें स्पष्ट रूप से बताएं और समस्या को हल करने के लिए आगे बढ़ें। यदि आपका उत्तर आपके द्वारा की गई किसी धारणा पर निर्भर करता है, तो धारणा को आपके paper में स्पष्ट रूप से बताया जाना चाहिए
- कुछ प्रश्नों में आपको code snippet दिए गए हैं। मान लें कि code snippet main program के अंदर लिखा है, सही header files include की गयी हैं, आदि, और इसलिए, code compile होता है।
- शुभकामनाएं!

Q1 [4 Marks] Consider the set of five numbers `S = {1, 3, 4, 7, 9}`.
Write down a function **`f()`** that returns a number drawn from **S** uniformly at random. In other words, a call to **`f()`** (which takes no arguments) should necessarily return some element of **S**, and each element must be returned with equal probability (which works out to ⅕ = 0.2). You can use any of `rand()`, `randuv()`, and `RAND_MAX` in your code. Assume that the random number generator has already been seeded; you should not set the seed inside **`f()`**.

Translation: Q1 [4 Marks] पाँच संख्याओं के set `S = {1, 3, 4, 7, 9}` को पढ़ें।
एक function **`f()`** लिखें जो S से uniformly और random तरीके से निकाली संख्या return करती है। दूसरे शब्दों में बोले तो, **`f()`** जभी call किया जाता है (जिसमें कोई arguments नहीं है) तभी आवश्यक रूप से **S** से एक element return करना चाहिए, और प्रत्येक element को समान संभावना के साथ return किया जाना चाहिए (जो ⅕ = 0.2 होता है। आप अपने code में rand(), randuv(), और RAND_MAX में से किसी का भी उपयोग कर सकते हैं। मान लें कि random number generator पहले ही seed किया है; आपको **`f()`** के अंदर seed set नहीं करना है।

```
#include<simplecpp>
int f() {
    Common line in Methods 1, 2, 3: int S[] = {1, 3, 4, 7, 9};

    Method 1
        int index = rand() % 5;  return S[index];

    Method 2
        int index = randuv(0,5);  return S[index];

    Method 3
        while(true) {
            int n = rand() % 10;   // Can be replaced with randuv
            for(int i = 0; i < 5; i++) {
                if(n == S[i]) return n;
            }    }

    Method 4
    double x = randuv(0, 1);
    if (x < 0.2) return 1;  if (x < 0.4) return 3;
    if (x < 0.6) return 4;  if (x < 0.8) return 7;
    return 9;

    /* There would be other ways to solve this. Please go through
    the code of the students and award full/partial marks
    accordingly */
}
main_program {
    srand(time(NULL));
    cout << f() << endl;
}
```

Q2. [5 marks] Given below is a snippet of a program that accepts a natural number **N** followed by **N** natural numbers from the user, and then stores them in an array **A**. The numbers are entered in any arbitrary order. The program intends to order the numbers such that all the even numbers appear first, followed by all the odd numbers. However, the even numbers and the odd numbers should be in the same sequence as in the original array **A**. **Write your code in the box given to achieve the required functionality as just described.**

Translation: Q2. [5 marks] नीचे एक program snippet दिया गया है जो user से एक number **N** लेता है और उसके बाद **N** natural numbers लेके, एक array **A** में store करता है. Numbers, arbitrary order में input किए गए है. Program का उद्देश्य numbers को फेरबदल करना है ताकि सभी even numbers पहले आएं, और उसके बाद सभी odd numbers आएं। हालाँकि, even numbers और odd numbers original array **A** के समान क्रम में होनी चाहिए। जैसा कि अभी बताया गया है, आवश्यक functionality को प्राप्त करने के लिए दिए गए box में अपना code लिखें

Example 1. For `N = 16` and `A = 2 5 3 6 9 4 17 8 18 14 16 33 34 12 5 24`
Output is: `2 6 4 8 18 14 16 34 12 24 5 3 9 17 33 5`

Example 2: For `N = 6` and `A = 7 2 8 9 5 4`
Output is: `2 8 4 7 9 5`

Example 3: For `N = 6` and `A = 7 13 11 9 5 23`
Output is: `7 13 11 9 5 23`

```
#include<simplecpp>
main_program {
    int A[100], N;
    cin >> N;
    for(int k = 0; k < N; k++) {
        cin >> A[k];
    }
    bool sorted = false;
    while(!sorted) {
        int i = -1, j = -1;

        // Write down your code in the box below, as many statements
        // as needed to achieve the required functionality
```

```
  sorted = true;
  for(int k = 0;  k < N - 1; k++){
       if(a[k] % 2 == 1 && a[k + 1] % 2 == 0){
            i = k;
            j = k + 1;
            sorted = false;
       }
  }

/*  Marks Distribution
   -  1 Mark: Correct loop
   -  1 Marks: Correct condition to identify odd,even or
      even,odd pattern
   -  1 Mark: Correct value set for variable i
   -  1 Mark: Correct value set for variable j
   -  1 Mark: Correct value set of bool sorted
*/
```

```
        if(i!=-1 && j!=-1) {
            int temp = A[i];
            A[i] = A[j];
            A[j] = temp;
        }
    } // End of while
    for(int k = 0; k < N; k++) {
        cout << A[k] << " ";
    }
} // End of main
```

Q3 [5 marks]

A 2D array of size 200 x 200 containing numbers from 0 to 255, both inclusive represents the pixel values of an image. A pixel value of 0 means that the pixel is black in colour. As the value increases, the shade of the grey colour becomes lighter, and eventually, in the end, the value 255 represents white i.e. the lightest shade of grey. (0 to 255 is black to white).

Thresholding is a technique that converts a grey-scale image into a black-and-white image. It is done by comparing each pixel of the image with a threshold value T. If the pixel value is greater than the threshold, assign the white color to it i.e. 255, or else assign black i.e. 0.

Your task is to write a C++ program as follows:

- In the main program,
  - accept 2D array of size 200 x 200 and a threshold T
  - Pass the array and the threshold value to a function called `thresholding`
  - Finally print the value of the updated 2D array
- In the `thresholding` function,
  - Check every pixel value of the 2D array and assign black or white as described above
  - Note that you cannot use another array and need to modify the same array

Write your solution on the next page

Translation: Q3 [5 marks]

200 x 200 size की एक 2D array जिसमें 0 से 255 तक की संख्याएँ शामिल हैं, दोनों समावेशी, एक image की pixel value को प्रतिनिधित्व करती हैं। 0 pixel value का अर्थ है कि pixel का रंग काला है। जैसे-जैसे value बढ़ता है, grey रंग की छाया हल्की हो जाती है, और अंततः, value 255 सफेद यानी grey की सबसे हल्की छाया का प्रतिनिधित्व करती है। (0 से 255 काला से सफेद है)।

Thresholding एक ऐसी तकनीक है जो grey scale image को black and white image में बनाती है। यह image की प्रत्येक pixel और threshold T के साथ तुलना की जाती है। यदि pixel value, threshold से अधिक है, तो उसे सफेद रंग यानी 255 assign करें, या फिर काला यानी 0 assign करें।

आपका task नीचे अनुसार एक C++ program लिखना है

- main program में
  - 2D array जिसकी size 200 x 200 है और एक threshold T को accept करे
  - Array और threshold को function में pass करें जिसका नाम thresholding है
  - अंत में updated 2D array की values को print करें
- Thresholding function में
  - 2D array की प्रत्येक pixel value की जांच करें और ऊपर बताए अनुसार काला या सफेद assign करें
  - ध्यान दें कि आप किसी अन्य array का उपयोग नहीं कर सकते हैं और आपको उसी array को modify करने की आवश्यकता है

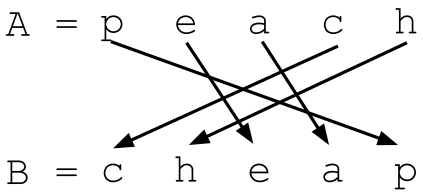
अपना जवाब अगले page पर लिखें

```
Marks Distribution
Main
      - 0.5 marks: Correct declaration of array and T
      - 0.5 marks: Correctly accepting elements of 200 x 200 array and
        the threshold T
      - 0.5 marks: correctly calling the function with arguments
      - 0.5 marks: correctly printing values of the array
Thresholding function
      - 1 mark: correct function signature (i.e. arguments)
      - 1 mark: correct condition
      - 1 mark: correct assigning of 0 or 255
```

```
#include <simplecpp>
void thresholding(int arr[][200], int T) {
    for (int i = 0; i < 200; i++) {
        for (int j = 0; j < 200; j++) {
            if (arr[i][j] > T) {
                arr[i][j] = 255; // White
            } else {
                arr[i][j] = 0;   // Black
            }
        }
    }
}
main_program {
    int T, image[200][200];
    for (int i = 0; i < 200; ++i) {
        for (int j = 0; j < 200; ++j) {
            cin >> image[i][j];
        }
    }
    cin >> T;
    thresholding(image, T);
    for (int i = 0; i < 200; i++) {
        for (int j = 0; j < 200; j++) {
            cout << image[i][j] << " ";
        }
        cout <<endl;
    }
}
```

Q4 [6 Marks] If words A and B are anagrams, it means that each letter of the alphabet occurs the same number of times in A and B. Some examples of anagrams are (peach and cheap), (listen and silent), (heart and earth), (part and trap), (secure and rescue), etc. Some examples of non-anagrams are (rocket and ticket), (table and label), (p**r**each and cheap), (part and art), etc.

Translation: Q4 [6 Marks]
यदि शब्द A और B anagram हैं, तो इसका मतलब है कि वर्णमाला का प्रत्येक अक्षर A और B में समान संख्या में आता है। कुछ anagram के उद्धरण है: (peach and cheap), (listen and silent), (heart and earth), (part and trap), (secure and rescue), etc. कुछ non-anagram के उद्धरण है: (rocket and ticket), (table and label), (p**r**each and cheap), (part and art), etc.



In the code snippet given below, in the main function, we accept two words from the user and store them in a 2D array that is then passed to the function **`isAnagram`**. Code up the function to find out whether the two words entered by the user are anagrams of each other. If yes, then return `true`, else return `false`. Assume that the user enters only valid lowercase letters (**`a to z`**), and the length of each word is less than 100.

Translation: नीचे दिए गए code snippet के, main function में, हम user से दो शब्द लेते हैं और उन्हें 2D array में store करते हैं जिसे फिर isAnagram function में pass किया जाता है। यह पता लगाने के लिए function को code करें कि user द्वारा input किए गए दो शब्द एक-दूसरे के anagram हैं या नहीं। यदि हां, तो true return करें, अन्यथा false return करें। मान लें कि user केवल valid lower case alphabets **(a to z)** input करता है और प्रत्येक शब्द की लंबाई 100 से कम है।

```
#include<simplecpp>
bool isAnagram(char words[][100]) {
    int count[256], len1 = 0, len2 = 0;
    for (int i = 0; i < 256; i++) {
        count[i] = 0;
    }
    len1 = strlen(words[0]);
    len2 = strlen(words[1]);
    if(len1 != len2)
        return false;
    for (int i = 0; i < len1; i++) {
        count[words[0][i]]++;
    }
    for (int i = 0; i < len2; i++) {
        count[words[1][i]]--;
    }
    for (int i = 0; i < 256; i++) {
        if (count[i] != 0)
            return false;
    }
    return true;

/*

There are multiple ways this can be coded. So, please check
the logic of the code written by the students and award
full/partial marks accordingly.

*/

}
main_program {
    char words[2][100];
    cin.getline(words[0], 100);
    cin.getline(words[1], 100);
    if (isAnagram(words))
        cout << "Yes" << endl;
    else
        cout << "No" << endl;
}
```

Q5 [5 marks] Go through the code snippet given below.
Translation: Q5 [5 marks] नीचे दिए गए code snippet को पढ़ें।

The main program accepts a non-negative integer **n** from the user and passes it to a recursive function **fun**. Code up the recursive function **fun** such that it prints the binary equivalent of the number, **n**, that was passed from the main function. Note that you cannot use any loops (repeat, while, do…while, for); **solutions using loops will automatically be given zero marks**. You need to print the binary number via recursion only.

Translation: Main program, user से एक non negative integer, n लेता है और इसे एक recursive function **fun** को pass करता है । Recursive function, **fun** को इस प्रकार code करें कि यह number, n, जो main function से ली गयी है, उसकी binary value print करनी चाहिए। ध्यान दें कि आप किसी भी loop का उपयोग नहीं कर सकते (repeat, while, do…while, for); solutions जो loop का इस्तेमाल करेंगे उनको खुद ब खुद 0 marks मिलेंगे। आपको binary number केवल recursion से print करना है।

E.g. n = 5, Output = 101
E.g. n = 9, Output = 1001
E.g. n = 38, Output = 100110

```
#include<simplecpp>
void fun(int n) {
```

```
Method 1
    if (n > 1) {
        fun(n/2);
    }
    cout << n % 2;
Method 2
void fun(int n) {
    if (n <= 1) {
        cout << n % 2;
    } else {
        fun(n/2);
        cout << n % 2;
    }
}
/* There are other ways to code this up. So, please check the
logic of the code written by the students */
```

```
}
main_program {
    int n;
    cin >> n;
    fun(n);
}
```

SPACE FOR ROUGH WORK

SPACE FOR ROUGH WORK