

**CS 101 (Autumn 2024)**

**Theory Quiz 2**

16 October 2024

Instructor: Manoj Prabhakaran

<b>Roll Number</b>	<b>SAMPLE</b>
<b>Division and Group</b>	
<b>Name</b>	

Q.No.	Marks / max	Graded By	Verified By	Re-grading (if requested)
1	/ 2			
2	/ 3			
3	/ 3			
4	/ 3			
5	/ 3			
TOTAL	/ 14			

**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!

**Question 1**

[2 marks]

Consider the code snippet given below.

**Rough work**

निचे दिए गए code snippet पर विचार करें।

```
////////////////////////////////////
void fun(int arr[], int n) {
    for (int i = 0; i < n; ++i) {
        arr[i] += i;
    }
}

int main() {
    int arr[100];
    int n;
    cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> arr[i];
    }
    fun(arr, n);
    for (int i = 0; i < n; ++i) {
        cout << arr[i] << " ";
    }
}
////////////////////////////////////
```

What is the output, if input is  
यदि input यह है, तो output क्या होगा?

5  
0 5 3 28 92

Ans: 0 6 5 31 96

// 1 mark: all numbers should  
be present in the same order  
// No partial marks

6  
10 65 23 8 56 78

Ans: 10 66 25 11 60 83

// 1 mark: all numbers should  
be present in the same order  
// No partial marks

## Question 2

[3 marks]

Consider the function `squeeze` below. The function has two arguments, `arr` and `n`, where `arr` is an array containing `n` non-negative integers in the range 0 to 10000.

The snippet intends to remove all the numbers that are divisible by 13, while shifting the subsequent numbers to the left. Append -1 in the empty slots at the end for every removed element.

E.g.: if `arr` is {7, 3, 9, 13, 12, 91, 94, 13, 26, 0, 17} when passed to the function, then after the function executes `arr` will be {7, 3, 9, 12, 94, 17, -1, -1, -1, -1, -1}.

ध्यान दें कि नीचे दी गई function 'squeeze' है। इस function में दो arguments हैं: 'arr' और 'n', जहां 'arr' एक array है जिसमें 0 से 10000 तक के 'n' non-negative integers शामिल हैं।

यह snippet सभी numbers को हटा देने का इरादा रखता है जो 13 से divisible होती हैं, जबकि इसके बाद की numbers को बाईं ओर shift करता है। हटा दिए गए element के लिए अंत में हर खाली स्थान में -1 जोड़ें।

उदाहरण के लिए: यदि 'arr' {7, 3, 9, 13, 12, 91, 94, 13, 26, 0, 17} है, जब इसे function में pass किया जाता है, तो function के execution के बाद 'arr' {7, 3, 9, 12, 94, 17, -1, -1, -1, -1, -1} होगा।

```
////////////////////////////////////
void squeeze(int arr[], int n) {
    int j=0;
    for(int i=0; i<n; i++)

        if (____ arr[i] % 13 != 0 ____ )

            ____ arr[j++] ____ = arr[i];

    while(j < n)

        ____ arr[j++] = -1 ____ ;
}
```

```
////////////////////////////////////
```

Fill in the blanks such that the function behaves as specified.

खाली स्थान भरें ताकि function निर्धारित के अनुसार behave करे।

## Rough work

Each blank carries 1 mark

- some students might have used a different approach. Please check and give marks appropriately.

- Multiple statements should not be accepted as an answer. Students should just fill in the blanks. However, let's say for the 3rd blank, if the students have written `arr[j] = -1;` and another statement `j--;`, in that case, give 0.5 marks for the blank instead of 1 mark. Similarly, for the second blank as well.

**Question 3**

[3 marks]

Consider the code snippet given below.

**Rough work**

निचे दिए गए code snippet पर विचार करें।

```
////////////////////////////////////  
int fun1(int n);  
int fun2(int n);  
  
int fun1(int n) {  
    if (n == 0) return 0;  
    if (n == 1) return 1;  
    return fun2(n - 1);  
}  
  
int fun2(int n) {  
    return fun1(n) + fun1(n - 1);  
}  
  
int main() {  
    int n;  
    cin >> n;  
    cout << fun1(n);  
}  
////////////////////////////////////
```

What is the output, if input is  
यदि input यह है, तो output क्या होगा?

Input: 5

Output: \_\_\_\_\_ **5** \_\_\_\_\_

Input: 8

Output: \_\_\_\_\_ **21** \_\_\_\_\_**1.5 marks each**

#### Question 4

[3 marks]

Consider the program given below for printing the factorial of the input. The function computing factorial is implemented as a *tail-recursive* function (i.e., the value returned by the recursive call is returned as it is). Fill in the two blanks in the function's body so that it works correctly on non-negative inputs.

नीचे दिए गए program पर विचार करें जो input का factorial print करता है। Factorial को compute करने वाला function एक tail-recursive function के रूप में implement किया गया है (यानी, recursive call द्वारा लौटाई गई value उसी तरह लौटाई जाती है)। Function के body में दो रिक्त स्थान भरें ताकि यह non-negative inputs पर सही तरीके से काम करे।

```
////////////////////////////////////
#include <iostream>
using std::cout; using std::cin;
using std::endl;

int fact(int n, int pdt = 1) {
    if(n==0)
        return pdt;
    return fact(n-1, n * pdt);
}

int main() {
    int n;
    cin >> n;
    cout << fact(n) //note: a single argument
        << endl;
}
////////////////////////////////////
```

#### Marking scheme - Blank 1

- 1.5 marks: correctly filled as shown
- 0.75 marks: `int pdt = 0 // i.e. incorrectly initialized`

#### Marking scheme - Blank 2

- 1.5 marks: correctly filled as shown
- 0.75 marks: `(n-1)*pdt` or something similar

**Rough work**

**Question 5****[3 marks]**

Consider the program given below, in which one of the expressions is written as EXP.

```
////////////////////////////////////
#include <iostream>
typedef double T;

// insert macro for defining EXP

int srch(T* begin, T* end, T t) {
    while(begin<end)
        if(*EXP == t)
            return end-begin;
    return -1;
}

int main() {
    T A[] = {2.7, 2.7, 1.0, 3.1, 2.7, 2.7};
    int N = 6;
    while(N>=0)
        std::cout << ( N = srch(A, A+N, 2.7) )
        << " ";
}
////////////////////////////////////
```

Consider two possible macros for defining EXP on the right. For each macro, state what the output of the resulting program will be. If the program results in undefined behaviour, you should explain why (instead of stating what the output will be).

दाईं ओर, 'EXP' को define करने के लिए दो possible macros पर विचार करें। प्रत्येक macro के लिए, बताएं कि resulting program का output क्या होगा। यदि program का परिणाम undefined behaviour होता है, तो आपको यह बताना चाहिए कि ऐसा क्यों है (output का वर्णन करने के बजाय)।

**Rough work**

```
#define EXP (--end)
```

Output: **5 4 1 0 -1**

**explanation is not required**

- 1.5 marks: **5 4 1 0 -1**
- 1 mark: **5 4 1 0**

```
#define EXP (end--)
```

Output: **Undefined behaviour**

**Explanation:** This macro uses the value of end before decrementing. In the first iteration, end is A+6, which is beyond the array bounds, and hence accessing \*end (and also \*(end--) ) causes undefined behaviour. If the program does not crash, it will likely output 4 3 0 -1.

● 1.5 marks: If the student mentions that there is undefined behaviour and provides a decent explanation. Output may or may not be written

● 1 mark: If the student mentions that there is undefined behaviour, does not explain the reason, but provides the output as 4 3 0 -1.

● 0.75 marks: If the student just writes the output as 4 3 0 -1 and does not mention undefined behaviour/explanation

**Rough work**

---

**Rough work**

---

**Rough work**