CS 101 (Autumn 2024)
Endsem

13 November 2024 - 08:30 AM to 11:00 AM

Instructor: Manoj Prabhakaran

Roll Number	SAMPLE
Division and Group	
Name	

Q.No.	**Marks** / **max**	**Graded By**	**Verified By**	**Re-grading** (if requested)
1	/ 5			
2	/ 5			
3	/ 5			
4	/ 5			
5	/ 5			
6	/ 5			
7	/ 6			
8	/ 8			
9	/ 8			
10	/ 8			
TOTAL	/ 60			

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 [5 marks]

Rough work

Consider the program given below.

नीचे दिए गए program पर विचार करें.

```
#include <iostream>
using std::cin; using std::cout;
int main() {
    int num;
    cin >> num;
    if (num % 3 == 0) {
        cout << "3";
        if( num % 5 == 0) {
            cout << "5";
        }
    }
    else {
        cout << "-1";
    }
}
```

What is the output if the input is as shown:

यदि input यह है, तो output क्या होगा?

Input: 9

Output: 3 // 1.25 marks.
// 0 marks if additional digits present.

Input: 15

Output: 35 // 2.5 marks
(1.25 marks for each digit.)
// OK if space added.
// 0 marks if additional digits present.

Input: 5

Output: -1 // 1.25 marks.
// 0.25 marks for writing 5.

Question 2 [5 marks]

Rough work

For each program below, write its output.

नीचे प्रत्येक program के लिए उसका output लिखें।

```
#include <iostream>
int main() {
  for(int i=0; i<10; ++i) {
    i += 2;
    std::cout << i << " ";
  }
}
```

Output: 2 5 8 11 // 2.5 marks

// 2 marks for "2 5 8".

// 1.5 marks for "2 4 6 8 10";

// 1 mark for other incorrect answers that start with 0 or 2, increment by 2 or 3, and don't go beyond 13.

```
#include <iostream>
int main() {
  int i=0;
  while(i<100) {
    i += 3;
  }
  std::cout << i << std::endl;
}
```

Output: 102 // 2.5 marks.
// 1.5 marks for 100, 101, 103.
// 1 mark for 97, 98, 99.

Question 3 [5 marks]

Consider the function given below which should return `true` if and only if its input is a prime number. There are four “blanks” in the code.

नीचे दिए गए function पर विचार करें जिसे `true` return करना चाहिए यदि और केवल यदि इसका input एक prime number है। Code में चार “blanks” हैं।

```
BLANK_1 isPrime(unsigned x) {
  if(x<2)
    return false;      // smallest prime is 2
  for(int i=2; i<x; ++i)
    if(BLANK_2 == 0)  // is i a factor of x?
       BLANK_3;
  BLANK_4;
}
```

Fill in the blanks below, so that the resulting function can be used to check if an unsigned integer is prime or not. (Assume that the macros given below will be inserted above the function.)

नीचे रिक्त स्थान भरें, ताकि परिणामी function का उपयोग यह जांचने के लिए किया जा सके कि कोई unsigned integer prime है या नहीं। (मान लें कि नीचे दिए गए macros को function के ऊपर लिखा जाएगा।)

```
#define BLANK_1    bool or int
// 1 mark

#define BLANK_2      x % i
// 1 mark.

#define BLANK_3   return false
// 1.5 marks. OK to add ";".
OK to write "return 0"
//(0.5 marks for return true)

#define BLANK_4   return true
// 1.5 marks. OK to add ";".
OK to write "return 1"
// (0.5 marks for return false)
```

Rough work

Question 4 [5 marks] Rough work

Consider the code snippet given below.

निचे दिए गए code snippet पर विचार करें।

```
#include <iostream>
using std::cin; using std::cout;
int main() {
    int arr[100];
    int n;                      // assuming n<=100
    cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> arr[i];          // read array elements
    }
    int sum = 0;
    for (int i = 0; i < n; ++i) {
        sum += arr[i];
        arr[i] = sum;           // update array elements
    }
    for (int i = 0; i < n; ++i) {
        cout << arr[i] << " "; // print array elements
    }
}
```

What is the output, if input is as shown:

यदि input यह है, तो output क्या होगा?

Input:

```
5
0 5 -5 5 -5
```

Output:

```
0 5 0 5 0                    2.5 marks: entire sequence is correct
0 0 5 0 5 0        1.5 marks: off-by-one (starts with 0 in 1st pos)
5 0 5 0            1.5 marks: off-by-one (skips the first position)
0 0 0 0 0                  1 mark: all positions have the final sum
0                       0.5 marks: only the final sum is printed once
```

Input:

```
4
10 1 1 1
```

Output:

```
10 11 12 13                  2.5 marks: entire sequence is correct
0 10 11 12 13      1.5 marks: off-by-one (starts with 0 in 1st pos)
11 12 13           1.5 marks: off-by-one (skips the first position)
13 13 13 13                1 mark: all positions have the final sum
13                      0.5 marks: only the final sum is printed once
```

Question 5

[5 marks]

Rough work

Consider the code snippet given below.

निचे दिए गए code snippet पर विचार करें।

```
#include <iostream>

int main() {
   int a = 1, b = 2;
   int* p = &a;
   int* q = &b;
   p = q;
   *p = (*q) + 1;
   std::cout << a << " " << b << std::endl;
}
```

What is the output of this program?

इस program का output क्या है?

Output: 1 3

2.5 marks for each number

2 marks total for "3 2"
(ignoring statement "p=q;") or
"2 2" (taking it as "q=p;")

Question 6 [5 marks] Rough work

Consider the program given below.

निचे दिए गए program पर विचार करें।

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

// checks if x,y,z are in sorted order (ascending or descending)
bool sorted(int x, int y, int z) {
  return (x<=y && y<=z) || (x>=y && y>=z);
}

// returns a reference to the element which is in the middle in sorted order
int& mid(int& a, int&b, int&c) {
  if(sorted(a,b,c)) return b;
  if(sorted(c,a,b)) return a;
  return c;  // in this case sorted(b,c,a)
}

int main() {
  int p, q, r;
  std::cin >> p >> q >> r;                        // reads 3 elements
  int& m = mid(p,q,r);                            // function returns a reference
  m = (p+q+r-m)/2;
  cout << p << " " << q << " " << r << endl;   // outputs 3 elements
}
```

For each input below, write down the output.

नीचे दिए गए प्रत्येक input के लिए, output लिखें।

Input: 10 6 24

Output: 15 6 24 // 2.5 marks (should be in the same order). (// 1.5 marks for "* 6 24". // 1 mark for having any two positions matching input.)

Input: 0 6 24

Output: 0 12 24 // 2.5 marks (should in the same order). (// 1.5 marks for "0 * 24". // 1 mark for having any two positions matching input.)

Question 7 [6 marks]

Consider the code snippet given below.

निचे दिए गए code पर विचार करें।

```
 1 #include <iostream>
 2 using std::cout; using std::cin;
 3
 4 struct node { int val; node* next; };
 5
 6 int main() {
 7   int n; cin >> n;
 8   node* head = nullptr;
 9   for(int i=0; i<n; ++i) { // To read n values
10     node* p  = new node;    // Allocate a new node
11     cin >> p->val;          // Read into the new node
12     p->next = head;         // Current head is attached as 'next' of the new node
13     head = p;               // The new node becomes head
14   }
15   while(head) {             // Print starting from head
16     cout << head->val << " ";
17     //
18     head = head->next;
19     //
20   }
21   cout << std::endl;
22 }
```

There are two parts to this problem.

इस problem के दो भाग हैं।

Part 1: What is the output of this program, on the following input?

Part 1: निम्नलिखित input पर इस program का output क्या है?

Input:
4
1 3 5 7

Output: 7 5 3 1
// 2 marks (order should match).
// 1 mark if order is reversed.

Part 2: Now, modify this program by inserting a statement each in line numbers 17 and 19 so that the output does not change, but when the control reaches line 21, all the memory allocated using `new` (in line 10) would have been freed.

Part 2: अब, line number 17 और 19 में से प्रत्येक में एक statement डालकर इस program को modify करें ताकि output में बदलाव न हो, लेकिन जब control line 21 तक पहुंचता है, तो `new` का उपयोग करके (line 10 में) allocate की गई सभी memory free हो जाएगी।

// Below t could be replaced with any identifier (not head)

Line 17: node* t = head; 2 Marks. OK to change spacing in "node * t"

Line 19: delete t; 2 Marks.

1 Mark total for "delete head" or "delete head->next" in either line.

Question 8 [8 marks]

The program given below reads two rational numbers (each as two integers – numerator followed by denominator) and prints them sorted in ascending order. If either denominator (variables `b,d` in the `main()` function) is 0, it prints an error message, using exception handling.

But there are six “blanks” in the code that you need to fill in, so that the resulting program works as described above.

```
#include <iostream>

// a function to check if p1/q1 < p2/q2
bool less(int p1, int q1, int p2, int q2) {
  if (q1==0 || q2==0) {
    std::string err =  "Division by zero";
    BLANK_1 err;
  }
  if(q1<0) { p1 = -p1; q1 = -q1; }
  if(q2<0) { p2 = -p2; q2 = -q2; }
  return p1*q2 < p2*q1;
}
// a function to sort two rational numbers
// in ascending order.
// uses the function above to compare them.
void sort(int& x1, int& y1,
          int& x2, int& y2) {
  if(less(BLANK_2)) {
    std::swap(BLANK_3);
    std::swap(BLANK_4);
  }
}

void printOrdered(int x1, int y1, int x2, int y2) {
  sort(x1,y1,x2,y2);
  std::cout << x1 << "/" << y1 << " "
            << x2 << "/" << y2 << std::endl;
}
int main() {
  int a, b, c, d;
  std::cin >> a >> b >> c >> d;
  BLANK_5 {
    printOrdered(a,b,c,d);
  } BLANK_6(std::string e) {
    std::cerr << e << std::endl;
  }
}
```

नीचे दिया गया program दो rational numbers को पढ़ता है (प्रत्येक दो integers के रूप में - numerator के बाद denominator) और उन्हें sorted ascending में print करता है। यदि कोई भी denominator (`main()` function में variables `b,d`) 0 है, तो यह excepition handling का उपयोग करके एक error message print करता है।

लेकिन code में छह “blanks” हैं जिन्हें आपको भरना होगा, ताकि परिणामी program ऊपर बताए अनुसार काम करे।

```
#define BLANK_1 throw
                // 1 mark


#define BLANK_2 x2, y2, x1, y1
                // 2 marks (1 mark
for "x1,y1,x2,y2")


#define BLANK_3 x1, x2
                // 1.5 marks.
// OK if exchanged with BLANK_4


#define BLANK_4 y1, y2
                // 1.5 marks.
// OK if exchanged with BLANK_3


#define BLANK_5 try
                // 1 mark


#define BLANK_6 catch
                // 1 mark
```

(Assume that the macros above will be inserted at the top of the program.)

Rough work

Question 9 [8 marks]

The *index* of a vector associates each element in the vector with a list of all the positions in which that element occurs. The index is organized so that elements appear in ascending order. For instance, a vector of 5 elements and its index are shown below:

किसी vector का *index* उस vector में प्रत्येक element को उन सभी positions की list के साथ जोड़ता है, जिनमें वह element आता है। index को इस तरह organize किया गया है कि elements ascending order में दिखाई दें। उदाहरण के लिए, 5 elements का एक vector और उसका index नीचे दिखाया गया है:

Vector:	Index:
`16, 49, 16, 27, 16`	`16: 0 2 4`
	`27: 3`
	`49: 1`

Consider the program below which accepts a number `n` and a vector of `n` integers, and prints its index.

नीचे दिए गए program पर विचार करें जो एक number `n` और `n` numbers की एक vector accept करता है, और इसके index को प्रिंट करता है।

```
#include <vector>
#include <map>
#include <iostream>
using std::cout; using std::cin;
using std::vector; using std::pair;

typedef pair<int,vector<int>> IndexEntry;
typedef vector<IndexEntry> Index;

Index mkIndex(const vector<int>& v);
void printIndex(const Index& ind);

int main() {
  vector<int> v;
  int n; cin >> n;
  for(int i=0; i<n; i++) {
    int x; cin >> x;
    v.push_back(x);
  }
  printIndex(mkIndex(v));
}
```

Note that the program declares two functions `mkIndex` and `printIndex` that need to be filled in.

ध्यान दें कि program दो function `mkIndex` और `printIndex` declare करता है जिन्हें भरने की आवश्यकता है।

```
Index mkIndex(const vector<int>& v) {
  std::map<int,vector<int>> M;
  // Update M so that for each item z in v
  // M[z] is a vector that contains all the
  // positions in v where z occurs.
  for(int i=0; i < v.size(); ++i)
    BLANK_1;

  // Now copy items in M into an Index
  Index ind;
  for(auto& x: M) // x is of type IndexEntry
    ind.push_back(x);
  return ind;
}
```

What shoud `BLANK_1` above be defined as?

```
#define BLANK_1 M[v[i]].push_back(i)
// 4 marks: 1 mark for M[ ],
1 mark for v[i], 1 mark for
.push_back(), 1 mark for i in
push_back(i)
```

```
void printIndex(const Index& ind) {
  for(auto& entry: ind) {
    cout << BLANK_2 << ": ";
    for(auto& i : entry.second)
      cout << BLANK_3 << " ";
    cout << std::endl;
  }
}
```

What should `BLANK_2` and `BLANK_3` above be defined as?

```
// 2 marks each

#define BLANK_2 entry.first

#define BLANK_3 i
```

Note: If `M` is a map, i.e., an object of class `std::map<keyT,valT>`, then `M[key]` returns a reference to the value stored for `key` (if none present, creates a default-initialized value first).

If `P` is an object of class `std::pair<T1,T2>`, then `P.first` and `P.second` are its two members.

Question 10 [8 marks]

The program given below defines a class to represent a point in 3D space. Study the code carefully and state what the output will be for the specified inputs.

```
#include <iostream>

class Point3D {
    double x, y, z;
public:
    Point3D(double x = 0, double y = 0, double z = 0) : x(x), y(y), z(z) {
        std::cout << "Constructed : (" << x << "," << y << "," << z <<")" << std::endl;
    }
    Point3D operator+(const Point3D& p) const {
        Point3D q(x+p.x, y+p.y, z+p.z);
        std::cout << "Operator + : (" << q.x << "," << q.y << "," << q.z <<")" << std::endl;
        return q;
    }
    Point3D& operator+=(const Point3D& p) {
        x += p.x; y += p.y; z += p.z;
        std::cout << "Operator += : (" << x << "," << y << "," << z <<")" << std::endl;
        return *this;
    }
    friend std::istream& operator>>(std::istream& is, Point3D& p) {
        is >> p.x >> p.y >> p.z;
        std::cout << "Read : (" << p.x << "," << p.y << "," << p.z <<")" << std::endl;
        return is;
    }
};

int main() {
    double a, b, c;
    Point3D p1;
    std::cin >> a >> b >> c >> p1;
    Point3D p2(a, b, c);
    p1 += (p1+p2);
}
```

For the input shown to the right, write down the output.

दाईं ओर दिए गए input के लिए, output लिखें।

`// If a particular output line is correct but not in this specified order, then deduct 50% marks for that output line.`

Input: `9 8 7 1 2 3`

Output:

```
Constructed : (0,0,0)
// 1 Mark

Read : (1,2,3)
// 1.5 Marks

Constructed : (9,8,7)
// 1 Mark

Constructed : (10,10,10)
// 1.5 Marks

Operator + : (10,10,10)
// 1.5 Marks

Operator += : (11,12,13)
// 1.5 Marks

OK to add/omit spaces. 50%
marks if only numbers are
wrong.
```

Rough work