Institute of Business Administration Karachi
Leadership and Ideas for Tomorrow

CSE141 Introduction to Programming (Fall'23)

Midterm Examination

IBA SMCS School of Mathematics and Computer Science

Max Marks: 70 *Time Allowed:* 2 hours

Name: ____________________ **ERP:** __________

Question:	Short Q/A	Tracing	Debugging	Coding	Total
Marks:	15	15	10	30	70
Score:					

Question 1: Short Q/A .... *15 marks*

(a) [9 marks]
For each of the expressions on the below, provide its C++ type (½ mark) and value (1 mark).
The used variables have been declared / initialized as shown on the right.
If a value cannot be determined, write "undefined" in the type column, and brief explanation in the value column.
The first two have been filled in as an example.

```
double x = 1.31;
double y = 20.98;
bool a = true;
bool b = false;
std::string s="mids";
```

Expression	Type	Value
`s + 4`	undefined	cannot use + for string and integer
`x + 5`	**double**	6.31
`(12 + 3)/ 4 * 2`	**int**	6
`x * (10/11)* y * 2`	**double**	0
`1 / 10 * d == d * 1 / 10`	undefined	undeclared variable d
`(!b || a)&& (!a || b)`	**bool**	**false**
`10.0 * (!a && !b)+ 20 * (!a || !b)`	**int**	20
`(int)(x + (int)(x + y))+ 0.1`	**double**	23.1

(b) [2 marks] Int the following code snippet, What value of `b` would cause the program to output as mentioned in the table below on the right? If there is none, type `none`.

```
char b = ________;
if (’q’ < b) {
    if (b > ’s’)
        cout << "R";
    else
        cout << "C";

    if (’s’ > b)
        cout << "U";
    else if (’s’ == b)
        cout << "T";
}
else if (b > ’q’)
    cout << "Z";
```

Value of `b`	Output
`none`	`"RT"`
`’r’`	`"CU"`
`’s’`	`"CT"`
`none`	`"Z"`

(c) [1 mark] Consider the following code-fragments below (right): which of the loops will give the same output?

```
// loop A
for(int i = 0; i < 5; ++i)
    std::cout << i << "\n";
```

```
// loop B
int i = 0;
while(i < 5) {
    std::cout << i << "\n";
    i++;
}
```

```
// loop C
int i = 0;
do {
    std::cout << i << "\n";
    i++;
} while (i < 5);
```

Solution: All three loops will give the same output.

(d) [3 marks] What do the following two code fragments do? Will these produces the same output? If not, what is the difference?

```
for (int i = 0; i < N; i++)
    for (int j = 0; j < N; j++)
        if (i != j) std::cout << i << ", " << j
                              << "\n";
```

```
for (int i = 0; i < N; i++)
    for (int j = 0; (i != j) && (j < N); j++)
        std::cout << i << ", " << j
                  << "\n";
```

Solution: First loop prints all pairs of numbers from `0` to `N-1` except the pairs where both numbers are same.

The second loop prints all pairs of numbers from `0` to `N-1` except the pairs where either both numbers are same or `j` is greater than `i`.

Question 2: Tracing .... *15 marks*

Find the output of the following programs.

(a)

```
#include<iostream>
using namespace std;
int main() {
    int i=5, j=7, k=3;
    if(i >= j && i >= k)
        cout << "Max = \n" << i;
    else if(j > i && j < k)
        cout << "Max = \n"<< j;
    else if(j > i && j > k)
        cout<<"Max = \n" << k;
    else
        cout<< "Error";
}
```

Solution:

```
Max =
3
```

(b)

```
#include<iostream>
using namespace std;
int main() {
    int i = 3;
    while(i) {
        int x = 2;
        for(int j = i; j < 5; j = j+x)
            cout << j << " ";
        i--;
    }
}
```

Solution:

```
3 2 4 1 3
```

(c)

```
#include<iostream>
using namespace std;
int main() {
    double x = 2;
    do {
        std::cout << x << " ";
        x /= 2;
    } while (x > 0.1);
}
```

Solution:

```
2 1 0.5 0.25 0.125
```

(d)

```
#include<iostream>
using namespace std;
int main() {
    bool x = false;
    bool y = false;
    while (!(x && y)) {
        cout << x << " " << y << " ";
        x = x || y;
        y = !y;
    }
}
```

Solution:

```
0 0 0 1 1 0
```

(e)

```
#include<iostream>
using namespace std;
int main() {
    int s = 0;
    int array[3][3] = {{0, 1, 2}, {3, 4,
        5}, {6, 7, 8}};
    for (int i = 0; i < 3 ; ++i) {
        for (int j = 2; j < 3 ; j++) {
            s += array[i][j];
        }
    }
    std::cout << s << "\n";
}
```

Solution:

```
15
```

Question 3: Debugging .... *10 marks*

(a) [5 marks] The following program attempts to examine an array and output whether all elements are positive. A flag named `allPos` is used. However, the Boolean logic is not implemented correctly in lines 11–16, so the program does not always output the correct answer. Find the problem and change lines 11–16, so that it will always output a correct result.

```
 1  #include <iostream>
 2  using namespace std;
 3  int main() {
 4      // initialize array with random numbers
 5      srand(time(0));
 6      int N=50; int x[N];
 7      for(int i=0; i<N; i++)
 8          x[i] = (rand()%99)-1;
 9
10      // check if all elements are positive
11      bool allPos = false;
12      for(int i=1; i<N; i++)
13          if(x[i]<0)
14              allPos = false;
15          else
16              allPos = true;
17
18      // output result
19      std::cout << boolalpha << allPos;
20  }
```

Solution: Lines 11–16 should be replaced with the following:

```
bool allPos = true;
for(int i=0; i<N; i++)
    if(x[i]<0)
        allPos = false;
```

(b) [5 marks] Complete the loop in following program to count number of positive and negative numbers separately in an array.

```
#include<iostream>
using namespace std;
int main() {
    int N = 12;
    int arr[] = {3, 5, -2, 9, -4, 10, -24, 19, 81, -7, 12, 13};
    int pcount=0, ncount=0;

    for(int i=0; i<N; i++) {

        if(arr[i]<0) { ncount++; }

        else     { pcount++; }
    }

    cout << pcount << " " << ncount;
}
```

Question 4: Coding .... *30 marks*

(a) [8 marks] Write a C++ program to calculate a dog's age in dog years. Note: For the first two years, a dog year is equal to 10.5 human years. After that, each dog year equals 4 human years.

Expected Output:

```
Input a dog’s age in human years: 15
The dog’s age in dog’s years is 73
```

```
#include<iostream>
using namespace std;
int main() {
```

Solution: Note: The statement of the problem and the example given contradicts each other. The statement should be corrected to say that: *For the first two years, a human year is equal to 10.5 dog years. After that, each human year equals 4 dog years.* The following code is based on this assumption. (But full marks will be awarded if your answer is based on the original statement.)

```
    cout << "Input a dog’s age in human years: ";
    double hy;
    cin >> hy;

    cout << "The dog’s age in dog’s years is ";
    if(hy<=2)
        cout << hy*10.5 << "\n";
    else
        cout << (hy-2)*4+21 << "\n";
```

```
}
```

(b) [10 marks] Write a program that take to find the sum of the following series

$$1 + \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}$$

where x and n are given as input.

```
#include<iostream>
using namespace std;
int main() {
    int x, n;
    cin >> x >> n;
```

Solution:

```
    double num=1, denom=1;

    double S=1;
    for(int i=1; i<=n; i++) {
        num *= x;
        denom *= i;
        S += num/denom;
    }

    cout << S;
```

```
}
```

(c) [12 marks] *Zero out matrix rows and columns:* Write a program that given an `m`-by-`n` integer matrix `a[m][n]`: if `a[i][j]` is `0`, set all elements in row `i` and column `j` to `0`.

Do not use any extra arrays.

For example, if the input is left matrix below, then the output should be the matrix on the right.

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 0 & 9 & 0 & 1 \end{bmatrix} \qquad\qquad \begin{bmatrix} 0 & 2 & 0 & 4 \\ 0 & 6 & 0 & 8 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Solution: *Using extra arrays:*

First, for every element `a[i][j]` that is `0`, set `row_flags[i]` and `col_flags[j]` to `true`. Next, set `a[i][j]` to `0` if either `row_flags[i]` or `col_flags[j]` is `true`.

```
#include<iostream>
int main() {
    const int m=3, n=4;
    int arr[m][n] = { {1, 2, 3, 4},
                      {5, 6, 7, 8},
                      {0, 9, 0, 1}};

    bool row_flags[m] {}; // init to false
    bool col_flags[n] {}; // init to false

    // store info in row_flags[] and col_flags[] arrays
    for(int i=0; i<m; i++) {
        for(int j=0; j<n; j++) {
            if(!arr[i][j]) { // if entry is 0
                row_flags[i] = true;
                col_flags[j] = true;
            }
        }
    }

    // set arr[i][j]=0 if either row_flag[i] or col_flag[j] is true
    for(int i=0; i<m; i++) {
        for(int j=0; j<n; j++) {
            if(row_flags[i] || col_flags[j])
                arr[i][j] = 0;
        }
    }

    // print modified M
    using std::cout;
    for(int i=0; i<m; i++) {
        for(int j=0; j<n; j++)
            cout << arr[i][j] << " ";
        cout << "\n";
    }
}
```

Without using extra array:

First, check whether row `0` has a `0` and whether column `0` has a `0`; record this information in two `bool` variables. Next, for each element `a[i][j]` that is `0`, set element `a[i][0]` and `a[0][j]` to `0`. Finally, set `a[i][j]` to `0` if either `a[i][0]` or `a[0][j]`.

```
#include<iostream>
int main() {
    const int m=3, n=4;
    int arr[m][n] = { {1, 2, 3, 4},
                      {5, 6, 7, 8},
                      {0, 9, 0, 1}};

    // check 0-th row
    bool zero_row = false;
    for(int j=0; j<n; j++)
        if(!arr[0][j])
            zero_row = true;

    // check 0-th col
    bool zero_col = false;
    for(int i=0; i<m; i++)
        if(!arr[i][0])
            zero_col = true;

    // if arr[i][j]=0, set both arr[i][0]=0 and arr[0][j]=0
    for(int i=1; i<m; i++) {
        for(int j=1; j<n; j++) {
            if(!arr[i][j]) { // if entry is 0
                arr[i][0] = 0;
                arr[0][j] = 0;
            }
        }
    }

    // set arr[i][j] to 0 if either arr[i][0]=0 or arr[0][j]=0
    for(int i=1; i<m; i++) {
        for(int j=1; j<n; j++) {
            if(!arr[i][0] || !arr[0][j])
                arr[i][j] = 0;
        }
    }

    if(zero_col)
        for(int i=0; i<m; i++)
            arr[i][0] = 0;
    if(zero_row)
        for(int j=0; j<n; j++)
            arr[0][j] = 0;

    // print modified M
    using std::cout;
    for(int i=0; i<m; i++) {
        for(int j=0; j<n; j++)
            cout << arr[i][j] << " ";
        cout << "\n";
    }
}
```