National University of Computer and Emerging Sciences, Lahore Campus

| | | | |
|---|---|---|---|
| Course Name: | Programming Fundamentals | Course Code: | CS |
| Program: | CS | Semester: | Fall 2019 |
| Duration: | 60 Minutes | Total Marks: | 35 |
| Paper Date: | 7-Nov-19 | Weight | 15 |
| Section: | ALL | Page(s): | 2 |
| Exam Type: | Sessional - II | | |

**Student : Name:**_____ **Roll No.**_____ **Section:**_____

Instruction/Notes: Solve the exam on this question paper. No rough sheets

**Problem 1 [10 pts]** Provide the output of the following C++ code.

```
void main()
{
	int m = 7, n=m/2+1;
	char ch = '+';
	for (int i = 0; i < n - 1; i++)
	{
		for (int j = 0; j < i; j++)
			cout << " ";
		for (int j = 0; j < m; j++){
			if (j == m - 1 || j == 0)
				cout << ch;
			else
				cout << " ";
		}
		m = m - 2;
		cout << endl;
	}
}
```

**Output:**

**Rough**

**Problem 2 [10 pts]** Give the output of the following code:

```
int check(float& a, float b, float c)
{
    float r = (a / b * 100.0 + 0.50);
    int s = (int) r;
    a = s / 100.0;
    if (a - c >= 0)
    return 1;
    return 0;
}
```

```
int main(){
  float dat[5] = {750,740,755,745,730};
  for(int i=0; i < 5; i++)
  {
    cout << i << ":";
    if( check(dat[i],1000,0.75) )
      cout << dat[i];
    cout << endl;
  }
}
```

**Output**:

**Rough**

**Problem 3 [15 pts]** Write a C++ function called mergeArrays. It should accept three integer arrays: A, B and C, and two integers m and n representing the sizes of the first two arrays. It can be assumed that the size of C is m+n. Arrays A and B contains integers that are already sorted in the ascending order (i.e. increasing order). Your function must combine all the numbers in A and B and store them C in such a way that C is also sorted in the ascending order. Here is an example to clarify the requirements. In this case, the sizes of A and B are 6 and 4 respectively.

| **A** | 0 | 4 | 5 | 5 | 7 | 9 |
|---|---|---|---|---|---|---|

| **B** | -1 | 0 | 3 | 6 |
|---|---|---|---|---|

| **C** | -1 | 0 | 0 | 3 | 4 | 5 | 5 | 6 | 7 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|

**Note:** You can use as many loops as you like, but this task should be accomplished *without* the use of **nested loops**. In particular, you should not apply any sorting algorithm to this problem. You cannot use or create any other function.

**Rough**