National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Object Oriented Programming	Course Code:	CS217
Degree Program:	BS (CS, SE, DS)	Semester:	Spring 2022
Exam Duration:	60 Minutes	Total Marks:	25
Paper Date:	24-March-2022	Weight	15
Section:	ALL	Page(s):	4
Exam Type:	Midterm-I		

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

Instruction/Notes: Attempt all questions. Answer in the space provided. **Answers written on rough sheet will not be marked**. Do not use pencil or red ink to answer the questions**.** In case of confusion or ambiguity make a reasonable assumption.

Question 1: **(Marks: 5+5+5)**

Part(a)

Identify the error (syntax/logical) in the following code. Mention the error and highlight the exact line having the error/throwing the exception. Rewrite the corrected code (rewrite only that part of the code that requires correction) and show the output of the corrected code.

```
class Color{
  int red;
  int green;
  int blue;
  Color();
  Color(int,int,int);
  void print();
};
Color::Color(){}
Color::Color(int r,int g,int b){
   red = r;
   green = g;
   blue = b;
}

void Color::print(){
   cout << red << '.' << green << '.' << blue ;
}

int main(){
   Color c1, c2(100,150,255);
   c1.print();
   c2.print();

   return 0;
}
```

Corrected Code:

```
#include <iostream>
using namespace std;
class Color{
private:
    int red;
    int blue;
    int green;
public:
    Color();
    Color(int,int,int);
    void print();
};
Color::Color()
{
    red=0;blue=0;green=0;
}
Color::Color(int r, int g,int b)
{
    red=r;
    blue=b;
    green=g;
}
void Color::print()
{
    cout<<red<<'.'<<green<<'.'<<blue;
}
int main()
{

    Color c1;
    Color c2(100,150,255);
    c1.print();
    c2.print();
    return 0;
}
```

	Output: **0.0.0100.150.255**

Part(b)

Identify the error(s) (syntax/logical) in the following code. Mention the error and highlight the exact line having the error/throwing the exception. Rewrite the corrected code (rewrite only that part of the code that requires correction) and show the output of the corrected code.

```
void AllocateMemory(int* arr)
{
	arr = new int[5];
}
void main()
{
	int* arr[3];
	int value = 1;

	for(int i=0; i<3 ; i++)
	{
		AllocateMemory(arr[i]);
		for(int j=0 ; j<5 ; j++)
		{
			arr[i][j] = value;

			value++;
		}
	}
	for(int i=0; i<3 ; i++)
	{
		for(int j=0; j<5 ; j++)
		{
			cout<<arr[i][j]<<"\t";
		}
		cout<<endl;
	}
}
```

Corrected Code:

```
void AllocateMemory(int*& arr)
{
	arr = new int[5];
}
int main()
{
	int* arr[3];
	int value = 1;

	for(int i=0; i<3 ; i++)
	{
		AllocateMemory(arr[i]);
		for(int j=0 ; j<5 ; j++)
		{
			arr[i][j] = value;

			value++;
		}
	}
	for(int i=0; i<3 ; i++)
	{
		for(int j=0; j<5 ; j++)
		{
			cout<<arr[i][j]<<"\t";
		}
		cout<<endl;
	}

    return 0;
}
```

Output:

```
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
```

Part(c)

What is the output of the following code

```
void fun(int* a,int s,int* f, int m){
    for(int i=0; i < s; i++){
        if (*(a+i) < m){
            (*(f + *(a+i)))++;
        }
    }
}

int main()
{
    int array[] = {2,3,2,2,1,7,3,4,0,1};
    int result[5] = {0};
    fun(array,10,result,5);
    for(int i=0;i < 5;i++){
        cout << i << ':' << result[i] << endl;
    }

    return 0;
}
```

Output:

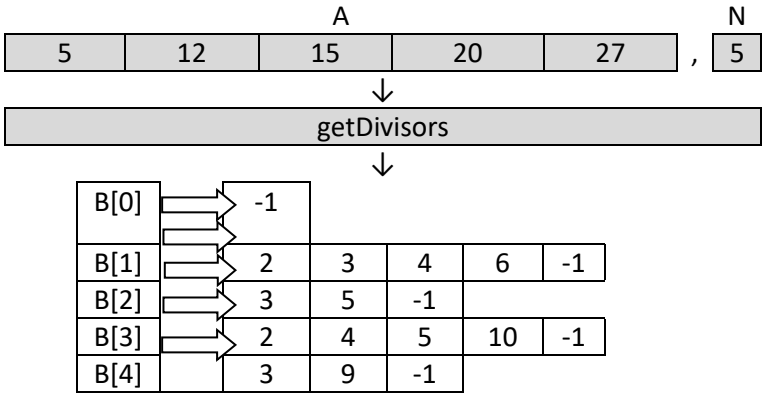
0:1
1:2
2:3
3:2
4:1

Question 2:

Write a C++ function ***getDivisors*** that receives an array, A, containing non-negative integers, and its size, n. The task is to compute the Divisors (other than 1 and the number itself) of all the numbers in A. The function must accomplish this task in the following way:

- All Divisors of an integer must be stored in a separate, dynamically allocated array, with -1 placed in the last index. The size of the dynamic array must exactly equal to #of Divisors+1.
- Pointers to these dynamic arrays are stored in another dynamic array (call it B) of size n. So that, when the function has finished, B[i] contains a pointer to the dynamic array containing the divisors of the number A[i], where $0 \leq i < n$.
- Lastly, the function returns B.

Following is an example input and its corresponding output, shown pictorially:



```
int ** getdivsior(int *A, int n)
{
  int **B;
  B=new int*[5];
int k; int count; int number;
   for (int i=0;i<n;i++)
   {
      count=0;
      number = A[i];
      k=2;
      while (k<number)
      {
         if (number%k==0)
         {
            count++;
         }
         k++;
      }
     B[i]=new int [count+1];
      k=2;

    for (int j=0;j<count+1;)
     {
         int temp=number%k;
         if (temp==0)
         {
            B[i][j]=k;
            j++;
         }
         k++;
     }
      B[i][count]=-1;

   }

   return B;
}
```