Computer Programming

Midterm-1, Fall 2016, Section (A, B, C)

Roll No.:_____ Section: _____

Date: September 20, 2016 **Total Marks: 40** **Total Time: 60 Minutes**

NOTE: Answer in the space provided. You can ask for rough sheets but they won't be graded

NOTE TO INVIGILATORS: Please do not collect the rough sheets handed out to the students

Question 1 [5+5+5+5+5 Marks]: Identify errors (Syntax, Logical, Memory Leak, or Dangling Pointers) in following pieces of code and provide Output where required.

Piece of Code	Output/Error

A)
`int main(){`
`  int * array[4];`
`  for (int i = 0; i < 4; i++)`
`  array[i] = new int[3];`
`  delete array;`
`  return 0;`
`}`	

B)
`int * Multiply(int * a, int * b){`
`  int s = *a * * b;`
`  return &s;`
`}`
 	
`int main(){`
`  int a= 10,  b = 5;`
`  int * result = Multiply(&a, &b);`
`  cout << *result << endl;`
`  return 0;`
`}`	

C)
 	
`char * stringCopy(char * str){`
 	
`  int len = strlen(str);`
`  char * ch = new char[len];`
`  for (int i = 0; i < len; i++)`
`  ch[i] = str[i];`
`  return ch;`
`}`
 	
`int main(){`
`  char a[] = "Happy:)";`
`  cout << stringCopy(a);`
 	
`  return 0;`
`}`	

Roll No.:_____ Section: _____

D)

```
void shrinkMe(int * array, int size){
    int * array2 = new int[size/2];
    for (int i = 0; i < size / 2; i++)
            array2[i] = array[i];
    delete[] array;
    array = array2;
}

int main(){
    int s = 10;
    int  * array = new int[s];
    for (int i = 0; i < s; i++)
            array[i] = i;

    shrinkMe(array, 10);

    for (int i = 0; i < s/2 ; i++)
       cout<< array[i] << endl;

    return 0;
}
```

E)

```
void DoSomething(int ** array, int size){

   int j = size - 1;
   for (int i = 0; i < size; i++, j--)
      (*array)[j] = (*array)[i];
}

int main(){
    int s = 5;
    int  * array = new int [s];
    for (int i = 0; i < s; i++)
       *(array + i) = i;

    DoSomething(&array, s);

    for (int i = 0; i < s; i++)
        cout << *(array + i) << endl;

     delete[] array;
     array = nullptr;

      return 0;
}
```

Question 2 [10 Marks]: Write a C++ function **`allPositionsOfSubarr`** which receives two integer arrays **arr** and **`subarr`** respectively. The goal of the function is to return all the positions in **`array`** where there is an occurrence of the sub array **`subarr`**.

For example, if **`arr`** is the {**`1,2,3,5,7,9,2,3,6,8,2,3,5,9`**}, and **subarr** is the {**`2,3`**} – then your function should return an array containing the numbers **`[1, 6, 10,-1]`**, as these are the indices in the array **`arr`** where an instance of the **subarr** begins and **`-1`** is the end marker.

Please note the following:

- The array that is being returned should be of size exactly one more than the number of times subarr occurs in **`arr`** (for example, it is exactly of size **`4`** (one more than the number of occurrences in the case shown above). In case **subarr** does not occur in **`arr`**, your function should return **`NULL`**.
- Make sure there are no memory leaks in your function.
- You don't need to take any inputs or do any outputs.
- Pay special attention to the function arguments and their types.
- You do not need to write the main function. Only the code for **`allPositionsOfSubarr`** is required.