[1*] 1. Write a loop that prints out the values **4**, **5**, **9**, **17**, and **12** without using an array or a **vector**.

[1*] 2. Given two **char***s pointing into an array, find and output the number of characters between the two pointed-to characters (zero if they point to the same element). Test your written function with the following

```
const char* s = “To be, or not to be, that is the question”;
const char* p1 = &s[21]; // point to 1st 't'
const char* p2 = &s[34]; // point to 'q'
```

Use structured programming constructs and pointer arithmetic. Do not use pre-built C functions for string manipulation.

[2*] 3. Define a **struct** with a member of each of the types **bool**, **char**, **int**, **long**, **long long**, **double**, and **long double**. Order the members so as to get the largest size of the **struct** and the smallest size of the **struct**.

[1*] 4. Copy all even non-zero elements of an **int[]** into a **vector<int>**. Use a pointer and **++** for the traversal.

[2*] 5. Write the iterative version of the binary search which we discussed at the last session. Like what we did already, write your code for C-Style arrays and C++ vectors.

[1*] 6. What is the output of the following code?

```
void f(int a)
{
  while (a--) {
    static int n = 0; // initialized once
    int x = 0; // initialized n times
    cout << "n == " << n++ << ", x == " << x++ << ´\n ´;
  }
}
int main()
{
  f(3);
}
```