1) Examine the following function assuming the contents of the array A are those shown below.

```
A  | 5  | 9  | 9  | 1  | 9  | 5  |
```

```c
int f1(int A[])
{
    int count = 6;
    for(int i=0; i < 6; i++){
        for(int j=0; j < 6; j++){
            if(j != i && A[i] == A[j]){
                count--;
                break;
            }
        }
    }
    return count;
}
```

a) List, in order, the value of j each time the `break` statement is executed. All spaces may not be used.

<table>
<thead>
<tr>
<th>j</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st execution of break</td>
</tr>
<tr>
<td>2nd execution of break</td>
</tr>
<tr>
<td>3rd execution of break</td>
</tr>
<tr>
<td>4th execution of break</td>
</tr>
<tr>
<td>5th execution of break</td>
</tr>
<tr>
<td>6th execution of break</td>
</tr>
</tbody>
</table>

b) What will the final value of count be?

c) Assuming the size of the array, A, is n, what is the big-O running time of this function (in terms of n)?
2) Examine the following program, called ‘arg_strings’. Assume it is run at the command line as:

```$ arg_strings 50 ee101 fights on!```

```int main(int argc, char *argv[]) {
    int j, n, ml, used=0;
    n = atoi(argv[1]);
    char *ptr = new char[n+1];

    for(int i=2; i < argc; i++){
        ml = strlen(argv[i]);
        if( (used + ml) < n){
            strcpy(ptr+used, argv[i]);
            used += ml;
            ptr[used++] = '\';
            ptr[used] = '\0';
        } else
            break;
    }
    cout << "Result is: " << endl;
    cout << ptr << endl;
    delete [] ptr; return 0;
} ```

a) Record ANY and ALL changes to the three variables shown below. Not all spaces may be needed for each variable.

<table>
<thead>
<tr>
<th>i</th>
<th>ml</th>
<th>used</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

b) What will be printed to the screen?