Assignment 7 on December 31

How we would have liked you to write the program given the hints if you are not so good in programming and your C code mostly does not work after typing

(Note that there are other simple ways of doing this problem not following the method in hint, or variations even following the hint; if you had done any of that, fine, but in whichever way you do, you should try to practice the same approach – understand the assignment, write step-by-step on paper in English the steps and logic, then try to transform into code part by part)

1. Read the assignment fully and well and understand
    a. I found several of you who have not read it fully and started coding
2. Write in English, step-by-step. Many of you did not, or did very sketchily.

```
Read n
Read n integers in array A
Print the n integers read in array A
Initialize array B as empty
For each integer in array A
        Check if it is in B
        If not in B, add to B
Print integers in B
```

Note the indentations. You want to write properly so that you can easily see what comes inside what

3. Now try writing some code for it. It does not have to be the full C code yet. Also note that as you go through the step, you realize the need for the variables, so declare them on the top.

```
int n, A[20], i, j, B[20], count;
scanf("%d", &n);
for (i=0; i<n; i++)
        scanf("%d", &A[i]);
for (i=0; i<n; i++)
        printf("%d ", A[i]);
printf("\n");
/* Have to start with an empty array to which you will add. So declare the array B
on top. Now how do you say it is "empty"? Just keep a count of the number of
elements in it, so empty means the count is 0. Keeping count is another variable,
so go declare it on top
*/
count = 0;
for (i=0; i<n; i++)
{
```

```
        check if A[i] is in B;
        If not in B, add to B and increment count
}
for (i=0; i<count; i++)
        printf("%d ", B[i]);
printf("\n");
```

4. Ok, but how to check if A[i] is in B? You just need to compare A[i] with each element of B, So another loop. How many elements are in B at any time? Equal to whatever the value of the variable count is! Initally no element, so count is 0 (already done)

```
for (j=0; j < count; j++)
        if (A[i] == B[j])
                /* at least one element matches, so A[i] is already in B, no need to
                   check further, so use the break statement to break out of for loop
                 */
                break;
/* So now you are here outside the for (j=0;….) loop. Now you can be here for 2
reasons. Either because (1) A[i] did not match any element in B and the loop ended
because j=count (loop is supposed to run till j < count), or because (2) it matched
one element in B, in which case the break statement was executed at which time j
was less than count (or it would not have gone into the loop).
For (1), you need to add A[i] to B, for (2) nothing is to be done. So check with an if
 */
If (j == count)
{
        B[count] = A[i];
        /* increase count as there is now one more element in B */
        count++;
}
```

5. Putting this together in the main code (removing the comments), you get your final code (then add int main() etc. etc. around it as usual, not shown here)

```
int n, A[20], i, j, B[20], count;
scanf("%d", &n);
for (i=0; i<n; i++)
        scanf("%d", &A[i]);
for (i=0; i< n; i++)
        printf("%d ", A[i]);
printf("\n");
count = 0;
for (i=0; i<n; i++)
{
        for (j=0; j < count; j++)
```

```
        if (A[i] == B[j])
                break;
        if (j == count)
        {
                B[count] = A[i];
                count++;
        }
 }
for (i=0; i<count; i++)
        printf("%d ", B[i]);
printf("\n");
```

6. Now type this quickly in codeblock. You do have to plan the time. But on a normal assignment day, it is ok if you spend 30 minutes doing this, it will take less time when you finally type your code and run it. There will still be mistakes, but less. And you will get better if you practice it more.