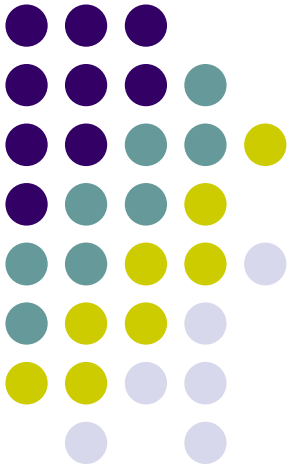
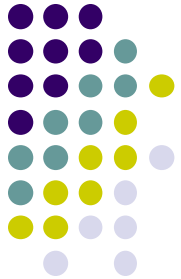
PDS Lab Section 15

December 31, 2020



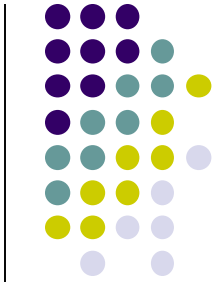


- Add header in front of your program

```
/*
*     Section 15
*     Roll No  :  20CS30010
*     Name     :  Your Name
*     Assignment No : 3
*     Description : Program to check points
*/
```

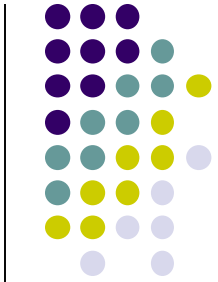
- Name your file with assgnX_Y.c, where X is the assignment number and Y is your roll no.
  - assgn7_20ME10010, assgn8_20CE30014,…..

Indenting your program



- Indentation - space in front of a line
  - Think of a paragraph beginning
- Give proper indentation in your program to make your program look nice
- Indent every block in your program consistently
  - What is a block? – statements inside if, else, for, while etc…(for now)
  - Idea is to be able to see which statements are part of which if or which else or which for/while etc. easily
    - Helps in making less mistakes in missing braces etc. when you write programs
    - Helps in easier debugging if your program does not work

Badly indented program

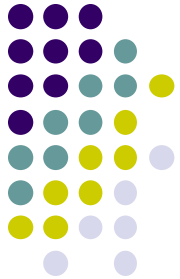


```
int main()
{
int i, j,k,n =10,sum, count;
for (i=0; i<n; i++)
{
sum=0; count=0;
for (j=0;j<n;j++)
{ scanf("%d",&k);
if (k>0)
{sum = sum + k;
 count = count + 1;
}
printf("Sum=%d Count=%d\n", sum, count);
}
}
}
```

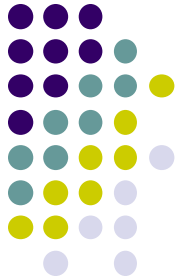
Properly Indented Program

```
int main()
{
    int i, j,k,n = 10, sum, count;
    for (i=0; i<n; i++)
    {
        sum=0;
        count = 0;
        for (j=0; j<n; j++)
        {
            scanf("%d",&k);
            if (k>0)
            {
                sum = sum + k;
                count = count + 1;
            }
            printf("Sum=%d Count=%d\n", sum, count);
        }
    }
}
```

Assignment Submission: NEW RULE FROM TODAY

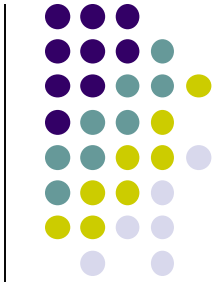


- For the first assignment, you will see two submission links in moodle
  - Assignment 7 Submission (Intermediate)
  - Assignment 7 Submission (Final)
- You **must submit** the .c file containing whatever you have done so far for Assignment 7 (first assignment today) in the link "Assignment 7 Submission (Intermediate)" strictly between 10-30 am to 10-45 am (**the link will open at 10-30 am and close at 10-45 am**)
  - Submit whatever you have done till then, we want to see how regularly you are progressing
  - **Not submitting will incur 30% penalty irrespective of what your final submitted version is**
- Submit the final .c file using the link "Assignment 7 Submission (Final)" as usual anytime before the lab ends
- For the second assignment (Assignment 8), there is a single submission link and submit as usual



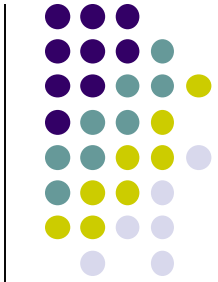
- What if you finish Assignment 7 before 10-30 and want to submit?
  - Wait till 10-30 and **submit the .c file in BOTH links**
    - **There must be a .c file submitted through both links (even if they are the same), or you lose 30% marks. Does not matter how good your final program is.**

- From now on, this is what you will need to do in every lab, so understand it well
  - If you have any confusion, ask us beforehand, **no excuses will be entertained later**

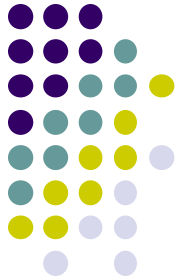


Assignments

Assignment 7

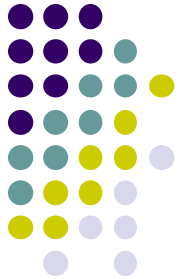


- Consider an array A of integers. In this assignment, you will have find and print all non-duplicate elements in A, in the same order as they appear in A
  - Ex:
    - If A has 12, 3, 4, 11,12, 4, 13, 12, you should print 12, 3, 4, 11, 13
    - If A has 3, 3, 3, 6, 1, 3, 3, 6, you should print 3, 6, 1
- Write a C program that
  - Read in an integer n (assume n < 20)
  - Read in n integers in an array A
  - Print the integers read
  - Print the non-duplicate integers in A as above



- Hint:
  - Use another array B, initially containing 0 elements
  - For each element in A (one loop), check if the element is already in B (compare with each element of B in another loop nested inside the first loop)
    - If yes, do nothing
    - If no, add to B after the last element (initially no element in B, so add to index 0 for the first one) and update the number of elements in B
  - At the end, print the elements in B

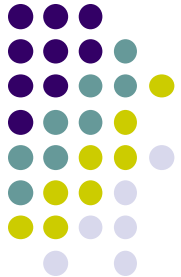
Assignment 8



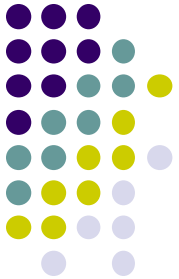
- A string is called a palindrome if it reads the same backward or forward
  - Example, “madam”, “radar”, “abcddcba”,….

Write a C program that will do the following in the order given:

- Reads in a string in a character array S using %s format in scanf
  - Type the string with no space and end it with pressing Enter
  - Assume that the size of the string (including the ‘\0’ at the end will not be more than 50 characters)
  - Remember that for reading with %s, you give only the name of the character array in which you want to read, without any & in front

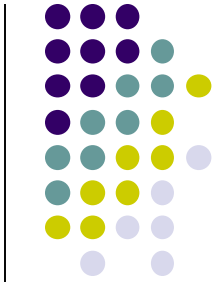


- Computes and prints the length of the string
  - Length = number of characters in the string not counting the ‘\0’at the end
  - Write a for loop to check the characters in the array S one by one till you get the character ‘\0’
- Reads in an integer p
- Checks if the string S has any palindrome of length p or not
  - If yes, print the palindrome part of the string only
    - If there are more than one such palindrome, print any one
    - Once you find such a palindrome, you should stop checking and break out of the loop (use break statement)
  - If no, print a message
- Do NOT use any other array (50% penalty if any other array is used)
- Hint: Start from each character in S from the start (one loop) and check if there is a palindrome of length p starting from it (another loop)



- Example:
  - If S = “abccba” and p = 6, prints “abccba” (as S is itself a palindrome of length 6)
  - If S = “abccba” and p = 4, prints “bccb” (as “bccb” is a palindrome of length 4 inside S)
  - If S = “abccba” and p = 2, prints “cc” (as “cc” is a palindrome of length 2 inside S)
  - If S = “abccba” and p = 3, prints “No palindrome of length 3 in S” (as there is no palindrome of length 3 inside S)
  - If S = “abbcddc” and p = 4, prints any one of “abbc” or “cddc” (as both are palindromes of length 4 inside S). Which one you hit first will depend on the way you write your program.
  - In all cases, once it finds a palindrome and prints it, it should not check for palindromes any more (break from for loop)

Teaser problems (not to be submitted)



- Read in a positive integer n (n < 100). Read in n integers (which may be positive, negative, or zero). Find the LARGEST and SMALLEST possible continuous window of sequence of the numbers (in the order read) whose sum is the smallest among all possible such sequences. For example if n = 8 and the numbers read are [3, ,-2, 2, -1, 1, -1, -4, 7], then the answer is [-2, 2, -1, 1, -1, -4] for Largest sequence and [-1. -4] for Smallest Sequence. Do not use any other array except the one to read in the numbers