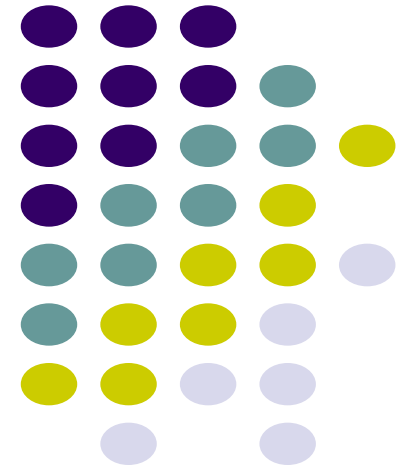
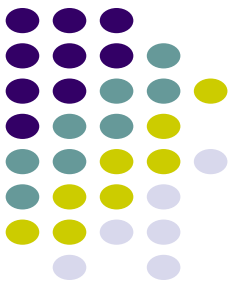


PDS Lab Section 15

December 10, 2020

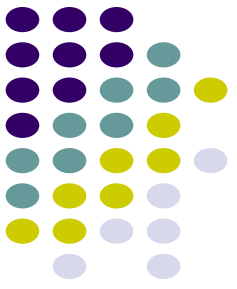


Grading Policy (Tentative)



- 40% marks will come from assignments
 - We will have around 10 assignment classes
 - Will take best 8 out of 10
- 40% marks will come from tests
 - We will have 3 tests (end-Dec/Jan-start, end-Jan/Feb-start, end-Feb/March-start) during regular class hours
 - Will take best 2 out of 3
 - The third test will be optional, you may or may not take it
- 20% marks will come from teachers' assessment
 - AG and PPC will interact with the students one-on-one over the semester
- All of the above is tentative, exact values/dates may change

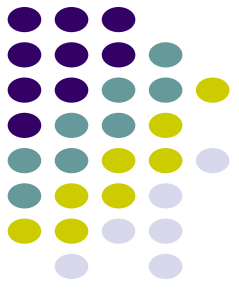
Your TAs



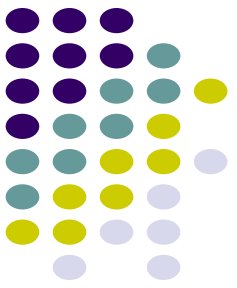
- Group 1: Nishkal Prakash (nishkalprakash@gmail.com)
- Group 2: Rohan Shankar (rohan9421837751@gmail.com)
- Group 3: Abhishek Khari (khari.abhishek1@gmail.com)
- Group 4: Arnab Mondal (mondal.arnab27@gmail.com)
- Group 5: Rajpratim Mitra (jonty.mitra@gmail.com)
- Group 6: Soumyadip Majumdar (soumyadipmajumdar668@gmail.com)
- Group 7: Narendra Parmar (workingnarendra@gmail.com)

Please know your group number and TA name. Also, while the TAs have been instructed to help you, please also remember that they are students with their own works, so ask for help only after trying yourself.

Remember before we start

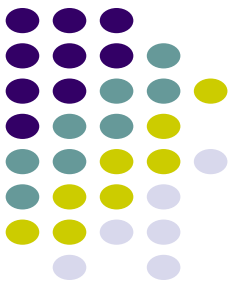


- If you get compilation error or your output is not correct
 - Debug the way we told you to in instructions
- Ask TAs/us only after you have tried yourself hard to solve your problem
 - No big hurry, you have > 1 hour for each assignment
 - Be cool, do not get tense or hurry to finish if you do not know programming well, you have enough time
 - You will get partial marks for your efforts even if it is not fully finished or correct.
- Submit .c file for Assignment 1 in link for Assignment 1, and .c file for Assignment 2 in link for Assignment 2 in moodle. Do not mess up.
 - Always submit whatever you can do
 - **Submission closes at 12-10 pm, so start submitting much before that**
 - **Server closes submission automatically at that time, so you will not be able to submit after that**
 - **Do NOT send any submission to us by email, we will ignore and not answer**

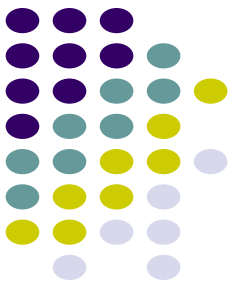


Recap of what you will need today

Input and Output



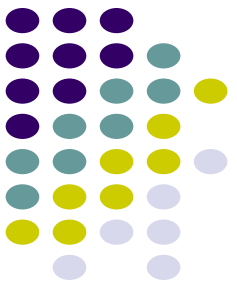
- Input from keyboard: *scanf* statement
 - Specify a format to indicate how to interpret the data, and *pointer to a variable* to store the data in
 - The type of the variable must match the corresponding format
- Output to monitor: *printf* statement
 - Specify a format to print the data
 - A variable where the data is stored
 - The type of the variable must match the corresponding format
 - printf can also just print strings (anything that is not preceded by the % inside the “ “)



- Some possible formats:
 - **%d** : to read/write **int** (integer)
 - **%f** : to read/write **float** (floating point numbers)
 - **%lf** : to read/write **double** (large floating point numbers)
 - **%c** : to read/write **char** (a single character)

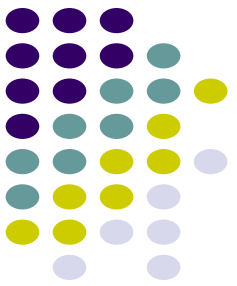
Examples

```
int main()
{
    int n;
    scanf("%d", &n);
    printf("The value of n read is %d \n", n);
}
```



IMPORTANT:

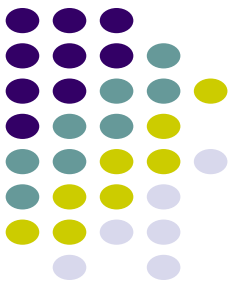
- **ALWAYS GIVE** the & before the variable name in **scanf**
- **NEVER GIVE** any & before the variable name in **printf**
- Until you know more, **do not put blank space or any other extra character within the quotes (“ ”) in scanf** other than %d, %f etc.



```
int main()
{
    float n;
    printf("Enter a floating point number: \n ");
    scanf("%f", &n);
    printf("The number %.2f is read \n", n);
}
```

If you input 23 from keyboard, this will print
"The number 23.00 is read"

Expression Evaluation



- Simple expressions as you know it with
 - $+$, $-$, $*$, $/$, $()$
- Assignment of the value of an expression to a variable

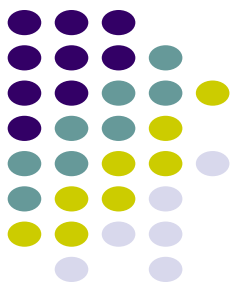
`int n;`

`n = 2*(3+5) - 4/2;`

`float f;`

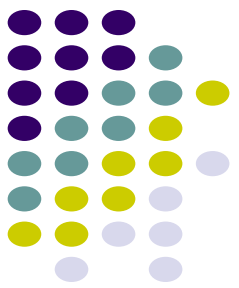
`f = 32.5/2.3*4;`

More complex expressions



- A whole bunch of mathematical functions are already defined that can be used
 - `sqrt`, `pow`, `cos`, `sin`, `acos`, `asin`, `log`, `log10`, `exp`,...
- Examples:
 - `sqrt(4)` will give 2
 - `pow(3,2)` will give 9
- You need to include `math.h` also
 - Add the line `#include <math.h>` just after the line `#include <stdio.h>`
- We will use `sqrt()` today

Simple program example



```
#include <stdio.h>
#include <math.h>
```

```
void main()
{
    int x1, y1, x2, y2;
    double len;
    scanf("%d%d%d%d", &x1, &y1, &x2, &y2);
    len = sqrt( (pow(x1-x2, 2)) + (pow(y1-y2, 2)) );
    printf("The line length is %f \n", len);
}
```

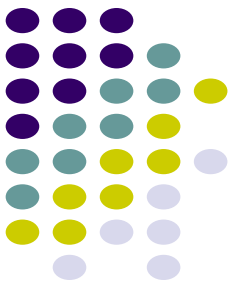
- Note that the value returned by `sqrt` is stored in variable of type `double` (this is true for most math functions)

Conditional Statement



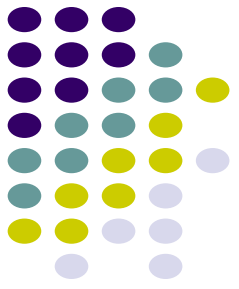
if (expression) statement1 else statement2;

- Executes **statement1** if expression evaluates to **true**, otherwise executes **statement2**
- May not have the **else** part
 if (expression) statement1;



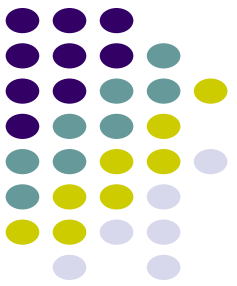
Expression types:

- Arithmetic: evaluates to **true** if $\neq 0$, else evaluates to **false**
 - **if (a – b) printf(“a not equal to b\n”);**
- Relational:
 - **if (a > b) printf(“a is larger than b\n”);**
 - Other operators: **<, ==, >=, <=, !=, ...**
- Boolean:
 - **if (a > b && a > c) printf(“a is greater than both b and c\n”);**
 - **&&** (logical AND), **||** (logical OR)

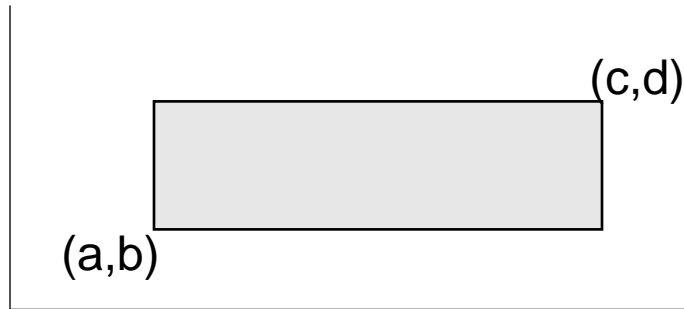


Assignments

Assignment 1

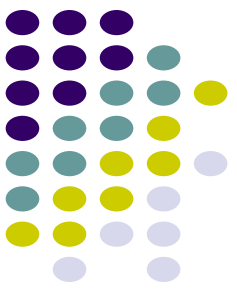


A rectangle with sides parallel to the X and Y-axis can be fully specified by specifying the (x,y) coordinates of the bottom-left corner (a, b) and the top-right corner (c, d) as shown below:



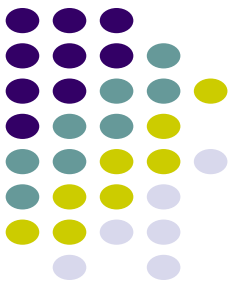
In this assignment, given a rectangle R specified as above and another point P with coordinates (p, q), you will find whether P is inside R or not (If P is on one of the boundaries of the rectangle, you should still consider it outside the rectangle).

For this assignment, assume that all coordinates will be integers.



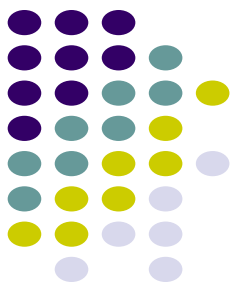
Write the program to do the following (exactly in this order):

- Read the coordinates (a, b) of the bottom-left corner of the rectangle (read both a and b in **a single scanf statement**)
- Read the coordinates (c, d) of the top-right corner of the rectangle (read both c and d in **a single scanf statement**)
- Compute the other two corners of the rectangle and store them in variables.
- Print all four corners of the rectangle nicely using **a single printf statement**. Print the corners starting from the bottom left corner (the point (a,b)) and going in the clock-wise direction (so bottom-left, then top-left, then top-right, then bottom-right). Your print should look something like **“The four corners of the rectangle are (...), (...), (...), (...)”** where the brackets will show the actual x,y coordinates of the corners
- Read in the coordinates (p,q) of the point to be checked using **a single scanf statement**.
- Check if the point is inside the rectangle or not and print out a suitable message like **“The point (...) is inside the rectangle”** or **“The point (...) is not inside the rectangle”**, where the (...) should show the actual x,y coordinates of the point

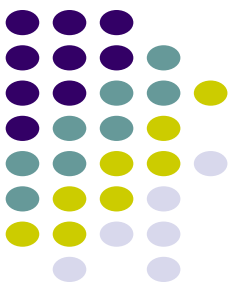


- Test your program with the rectangle specified by bottom left corner $(2,1)$ and top right corner $(7,5)$ for the points $(4,3)$, $(8,2)$, $(5,8)$, and $(3,7)$

Assignment 2



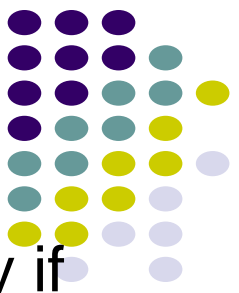
- Read in the coefficients a , b , c of the quadratic equation $ax^2 + bx + c = 0$. Assume that a , b , c are real numbers. Then
 - Print the values of a , b , and c read with the message “The coefficients of the equation are”
 - If the roots are complex, print a message saying so, nothing else to be done
 - Else, compute and print the roots of the equation with a message “The roots of the equation are...”. In addition, if the roots are equal, also print a message after printing the roots “The roots are equal”
- Test with the following sets of data
 - $a = 16.0$, $b = 24$, $c = 9$
 - $a = 4$, $b = 10$, $c = 4$
 - $a = 4$, $b = 5$, $c = 4$



- To do this assignment, you will need to also find square root. C provides a whole set of mathematical functions including square root (you will see other examples in theory)
- Here's a small sample C statements to find square root of a real number x

```
double s;  
s = sqrt(x);
```
- You may get a warning about x being not double if you have declared it as a float, ignore it for now, we will take care of it later

Teaser Problem (Not to be submitted)



- To be done only after you have finished the first two assignments, and only if you know how to program already
- Do it on your own, do not ask for help from TAs, they will be busy helping other students with the mandatory assignments

Let there be 3 rectangles having areas A_1 , A_2 and A_3 , where each area value is a positive integer. Find one set of three pairs of possible rectangle dimensions (L_1, B_1) , (L_2, B_2) , (L_3, B_3) where $L_1, L_2, L_3, B_1, B_2, B_3$ are ALL distinct integers (no two of them are the same) and $L_1 \cdot B_1 = A_1$, $L_2 \cdot B_2 = A_2$ and $L_3 \cdot B_3 = A_3$. Your program should read in A_1, A_2, A_3 as input from the keyboard. If such a combination does not exist the program should print "Not Possible", otherwise it should print the three pairs of dimensions. Note that A_1, A_2 and A_3 need not be distinct.

As a second part, find all such sets of three pairs of possible rectangle dimensions.