
Programming Assignment 5

- This programming assignment consists of 3 problems.
- It is due on Wednesday Nov. 30 in the lab by **5 pm**.
- *Lab structure and regulations:*
 - ★ The 3 hours Lab session is on Wednesdays in Lab rooms 1 and 2 from 2:00 pm to 5:00 pm. It consists of three parts:
 - *Occasional Solving Session (not graded but attendance mandatory)*
 - *Programming Assignment (graded)*
Programming Assignments will be posted on Moodle on weekly basis. Typically, a Programming Assignment requires much more than the time allocated for this part in the Lab, so you are supposed to complete the major part of the assignment at home. The Lab instructor will grade your assignment and can help you with the problems you are facing.
 - *Occasional graded weekly quiz*
 - ★ You are supposed to submit your own work. Cheating will not be tolerated and will be dealt with severely: zero grades on the programming assignments, disciplinary committee, Dean's warning.
 - ★ Lab attendance is mandatory. Violating this rule can lead to a failing grade.

Problem 1. (Circles: functions warmup)

The following formula gives the distance between two points (x_1, y_1) and (x_2, y_2) in the Cartesian plane:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Given the center and a point on the circle, you can use this formula to find the radius of the circle. Write a program that prompts the user to enter the center and a point on the circle. The program should then output the circle's radius, diameter, circumference, and area. Your programs must have at least the following functions:

- a) *distance*: This function takes as its parameters four numbers that represent two points in the plane and returns the distance between them.
- b) *radius*: This function takes as its parameters four numbers that represent the center and a point on the circle, calls the function distance to find the radius of the circle, and returns the circle's radius.

- c) *circumference*: This function takes as its parameter a number that represents the radius of the circle and returns the circle's circumference. (If r is the radius, the circumference is $2\pi r$.)
- d) *area*: This function takes as its parameter a number that represents the radius of the circle and returns the circle's area. (If r is the radius, the area is πr^2 .)

Assume that $\pi = 3.1416$.

Problem 2. (Selection Sort revisited)

- a) **Array Print function**: Write the function *arrayPrint* which takes as its parameters (a pointer to) an array of integers, and its size. It prints the array.
- b) **Array Smallest function**: Write a function *arraySmallest* which takes as parameters (a pointer to) an array A of integers and two indices *start* and *end* of A (assume $0 \leq start \leq end$). It is supposed to find and return the index of the smallest element in $A[start...end]$.
- c) **Selection Sort function**: Using the *ArraySmallest* function in part (b) and the *swap* function we did in class, write the selection sort function *selectionSort* which takes as its parameters (a pointer to) an array of integers, and its size. This function sorts the array using the Selection Sort algorithm [Problem 3 of Programming Assignment 3].

Using those functions, you are supposed to get a very simple and structured code for the selection sort algorithm.

- d) **Test Program**: Write the following program to test the above functions.

```

prototype of printArray
prototype of arraySmallest
prototype of swap
prototype of selectionSort

```

main function:

1. ask the user to enter an integer n
 2. allocate memory for an array A of integers of size n
 3. fill the array with the user input
 4. call the *printArray* function to print A
 5. call the *selectionSort* function to sort A
 6. call the *printArray* function to print the sorted version of A
 7. free the memory allocated for the array A
- end of main

bodies of the above 4 functions

Problem 3. (Pointers and reference parameters practice)

Write the functions:

a) *divide*

This function takes as input parameters two integers a and b . It returns via reference parameters the quotient q and the remainder r resulting from the division of a by b (i.e., q and r are nonnegative integers such that $a = qb + r$ and $r < b$).

b) *dividePtrsVersion*

Same as (a) but use pointers instead of reference parameters.

c) *rotateThree*

This function takes three integer variables by reference. It is supposed to rotate the content of the variables. For example, the following test program

```
int a = 2;
int b = 5;
int c = 7;
rotateThree(a,b,c);
cout<<a<<' ' <<b<<' ' <<c;
```

should output

```
7 2 5
```

d) *rotateThreePtrsVersion*

Same as (c) but use pointers instead of reference parameters.

Write a program to test your functions.