CS 2123 Data Structures Recitation - Recitation Exercise 02

Write a C program to work on pointers. In `main()` you will first

- Declare five integer variables named as `a, b, c, temp, sum` and set the initial values for `a, b, c,` as 10, 20, 30, respectively.
- Declare three integer **pointers** named as `pa, pb, pc.` These pointers should initially hold the addresses of variables `a, b, c,` respectively.
- Declare three integer pointers to pointers as `ppa, ppb, ppc.` These pointers to pointers should initially hold the addresses of pointers `pa, pb, pc`, respectively.
- If needed for the below tasks, you can declare other variables as well as pointers later..

- Now implement the following tasks in the `main ()`
  - Write the necessary statements to print the values of `a, b, c` by just using `a, pb, ppc` (direct, indirect, indirect indirect access)
  - Write the necessary statements to find the `sum` of the values in `a, b, c` by just using `a, pb, ppc, sum.` Then print `sum.`
  - Write the necessary statements to exchange/print the values in `a` and `b`, by just using `pa, ppb` and maybe `temp.` Then print `a` and `b.`
  - Write the necessary statements to exchange the values/addresses in `pa` and `pb` by just using `pa, ppb` and maybe a temp pointer that you can declare. Then print the values of `a` and `b` by directly using `a` and `b` as wells as indirectly by using `*pa` and `*pb.` Explain what happens here, why the values of `a` and `b` did not change when we access them directly while it seems the changed when we access them indirectly. Type your explanation as a C comment in your program after the printf statements.

- Now copy/paste the following functions before your `main()`

```
void increaseA(int val)   { val++;     }
void increaseB(int *pval) { pval++;    }
void increaseC(int *pval) { *pval++;  }
void increaseD(int *pval) { (*pval)++;  }
```

  Then call each of these functions in `main()` as follows

```
increaseA(a);
increaseB(&a);
increaseC(&a);
increaseD(&a);
```

  and print the value of `a` after each call to see what happens. Accordingly, explain why the first three call did not cause any change on variable `a` while the last one increases it by 1. Type your explanation as a C comment next to the above calls in your program.

- Finally implement a function exchange that "swaps pointer variables" as defined in slides 42 in ch02-datatypes-pointer-.... (erercise-02-pointer-swap) and call it as
  ```
  exchange(&pa, &pb);
  exchange(&pa, ppb);
  exchange(ppa, ppb);
  ```
  and print the values/addresses in pa and pb as well as the values at these addresses. Again explain which ones are working to exchange pa and pb. Why/why not!

- You can try other interesting things to do more practice with pointers... Good luck !

/* Don't forget to include comments about the problem, yourself and each major step in your program! */

What to return: !!!! NO LATE RECITATION ASSIGNMNET WILL BE ACCEPTED !!!

1. Create a directory called LASTNAME_Recitation02 and do all your work under that directory
2. First implement your program which can be named as rec02.c
3. Then compile and run it. Copy/paste the result in an output file, say out02.txt.
3. Finally zip your LASTNAME_Recitation02 directory as a single file LASNAME_Recitation02.zip and Go to BB Learn to submit it as **attachment** before the deadline.

You must submit your work using Blackboard Learn and respect the following rules:

1) All assignments must be submitted as either a zip or tar archive file unless it is a single pdf file.
2) Assignments must include all source code.
3) Assignments must include an output.txt file which demonstrates the final test output run by the student.
4) If your assignment does not run/compile, the output.txt file should include an explanation of what was accomplished, what the error message was that prevented the student from finishing the assignment and what the student BELIEVES to be the underlying cause of the error.
