COMP 322/L—Introduction to Operating Systems and System Architecture
Assignment #1--Process Creation Hierarchy

Objective:

To simulate process creation and destruction when implemented with a dynamic array.

Specification:

The program creates/destroys child processes based on choosing from a menu of choices, where each choice calls the appropriate procedure, where the choices are:

1) Enter parameters
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Assignment:

- Create a process creation hierarchy as a dynamic array of length n which references the process control blocks (PCBs), indexed 0 to n-1
- Each PCB is a structure consisting of four fields:
  - ***parent:*** a PCB index corresponding to the current process' creator
  - ***first_child:*** the first child process (if any) of the current process
  - ***older_sibling:*** a process (if any) that was created immediately before the current process and shares the same parent
  - ***younger_sibling:*** a process (if any) that was created immediately after the current process and shares the same parent
- The necessary functions are simplified as follows:
  - ***create()*** represents the create function, which prompts for the parent process PCB[p]. The function creates a new child process PCB[q] of process PCB[p] by performing the following tasks:
    - record the parent's index, ***p***, in PCB[***q***]
    - initialize the ***first_child*** and ***younger_child*** fields of PCB[***q***] to -1 (empty)
    - record the ***older_sibling*** field of PCB[***q***] as ***t***, where ***t*** is the located older sibling of ***q*** (if any)
    - record the ***younger_sibling*** field of PCB[***t***] (if it exists) as ***q***, where ***t*** is the located older sibling of ***q*** (if any)
  - ***destroy()*** represents the destroy function, which prompts for the parent process PCB[p]. The function recursively destroys all descendent processes (child, grandchild, etc.) of process PCB[p] by performing the following tasks:
    - the initially call the recursive procedure on the ***first_child*** field of PCB[***p***]
    - call the recursive procedure on the younger sibling of PCB[***q***] (this continues until the youngest sibling is reached)
    - call the recursive procedure on each younger sibling's ***first_child*** field
    - set all fields of the current process to -1 (***parent***, ***first_child***, ***younger_sibling***, ***older_sibling***)

What NOT to do (any violation will result in an automatic score of 0 on the assignment):

- Do NOT modify the choice values (1,2,3,4) or input characters and then try to convert them to integers--the test script used for grading your assignment will not work correctly.
- Do NOT turn in an alternative version of the assignment downloaded from the Internet (coursehero, chegg, reddit, github, etc.) or submitted from you or another student from a previous semester.
- Do NOT turn in your assignment coded in another programming language (C++, C#, Java).

What to turn in:

- The source code as a C file uploaded to Canvas by the deadline of 11:59pm PST (-20% per consecutive day for late submissions, up to the 4th day—note 1 minute late counts as a day late, 1 day and 1 minute late counts as 2 days late, etc.)
- Make sure your code compiles with the online C compiler before submitting: https://www.onlinegdb.com/online_c_compiler

Sample output

```
Process creation and destruction
--------------------------------
1) Enter parameters
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 1
Enter maximum number of processes: 5

Process creation and destruction
--------------------------------
1) Enter parameters
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 2
Enter the parent process index: 0
cr[0]:  creates 1st child of PCB[0] at PCB[1]

i       Parent  First   Older   Younger
-------------------------------------
0       0       1
1       0

Process creation and destruction
--------------------------------
1) Enter parameters
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 2
Enter the parent process index: 0
cr[0]:  creates 2nd child of PCB[0] at PCB[2]

i       Parent  First   Older   Younger
-------------------------------------
0       0       1
1       0                       2
2       0               1

Process creation and destruction
--------------------------------
1) Enter parameters
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 2
Enter the parent process index: 2
cr[2]:  creates 1st child of PCB[2] at PCB[3]

i       Parent  First   Older   Younger
-------------------------------------
0       0       1
1       0                       2
2       0       3       1
3       2
```

```
Process creation and destruction
--------------------------------
1) Enter parameters
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 2
Enter the parent process index: 0
cr[0]:  creates 3rd child of PCB[0] at PCB[4]

i       Parent  First   Older   Younger
-------------------------------------
0       0       1
1       0                       2
2       0       3       1       4
3       2
4       0               2

Process creation and destruction
--------------------------------
1) Enter parameters
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 3
Enter the process whose descendants are to be destroyed: 0
de[0]:  destroys all descendants of PCB[0] which includes:
PCB[4] PCB[3] PCB[2] PCB[1]

i       Parent  First   Older   Younger
-------------------------------------
0       0

Process creation and destruction
--------------------------------
1) Enter parameters
2) Create a new child process
3) Destroy all descendants of a process
4) Quit program and free memory

Enter selection: 4
Quitting program
```