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 linked lists.

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 two fields:
  - parent: a PCB index corresponding to the process' creator
  - children: a pointer to a linked list, where each node contains the PCB index of one child process and a link to the next child in the list

- 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:
    - allocate a free PCB[q]
    - record the parent's index, p, in PCB[q]
    - initialize the list of children of PCB[q] as empty (NULL)
    - create a new link containing the child's index q and append the link to the linked list of PCB[p]
  - 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: for each element q on the linked list of children of PCB[p]:
    - destroy(q) /* recursively destroy all descendants */
    - free PCB[q]
    - deallocate the element q from the linked list

What NOT to do:

- 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 alternate version of the assignment downloaded from the Internet (coursehero, chegg, reddit, github, etc.) or submitted from you or another student from a previous semester—the test cases from this semester will not work on a previous semester's assignment.
- 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.)
- As a note, even though your code may compile on a compiler you have installed on your computer, I do not have access to your computer. I will be using the following free online compiler for testing, so make sure your code compiles with the following 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
PCB[0] is the parent of: PCB[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: 0
PCB[0] is the parent of: PCB[1]  PCB[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: 2
PCB[0] is the parent of: PCB[1]  PCB[2]
PCB[2] is the parent of: PCB[3]

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
PCB[0] is the parent of: PCB[1]  PCB[2]  PCB[4]
PCB[2] is the parent of: PCB[3]

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 index of the process whose descendants are to be destroyed: 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...
```