Breaking down: The Game ofRocks!1.0 - 12 points

Answer the questions provided for each of the required functions in the Rocks 1.0 program

List of Required Functions:

genPile

```
/*generates and returns a random number between 15 and 30, inclusive
*/
int genPile()
```

What TYPE of function is this?
int function.
Does it RETURN anything? If so, what is the significance of the returned value?
Yes, it returns an int, it will generate and return a random number between 15 and 30.
How many PARAMETERS does this function accept?
0
List any parameters that are PASSED BY VALUE to this function:

- x

List any parameters that are PASSED BY REFERENCE to this function:

- x

adjustPile

```
/*subtracts number of rocks (remove) from pile's total (pile)
if the all rocks are gone, display a message that the pile
is empty and set the lose variable equal to true to indicate the
game has been lost
Parameters:
lose - has the game been lost?
pile - total number of rocks left in the pile
remove - the desired amount of rocks to remove in this turn
*/
void adjustPile(bool &lose, int &pile, int remove)
```

What TYPE of function is this?
void
Does it RETURN anything? If so, what is the significance of the returned value?
NO
How many PARAMETERS does this function accept?
3
List any parameters that are PASSED BY VALUE to this function:

- remove

List any parameters that are PASSED BY REFERENCE to this function:

- lose

- pile

compDecision

```
/*Computer determines and returns number of rocks to take,
based on following rules:
1 or 2 rocks: take 1
3 rocks: take 2
4 rocks: take 3
more than 4 - take random number 1-3
and returns a value representing the number of rocks to remove (do not adjust the
value inside pile here!)
Parameter:
pile – total number of rocks remaining in the current pile being played
*/
int compDecision(int pile)
```

What TYPE of function is this?
int
Does it RETURN anything? If so, what is the significance of the returned value?
Yes, it returns an int, returns the number of rocks to take based on the rules.
How many PARAMETERS does this function accept?
1
List any parameters that are PASSED BY VALUE to this function:

- pile

List any parameters that are PASSED BY REFERENCE to this function:

-

valid

```
/*Determines if the number of rocks indicated
is valid for the pile (i.e. not too many,
not too few. Returns true if the selected number IS valid
Parameters:
remove – # of rocks the user is trying to remove
pile – total number of rocks remaining in the pile
*/
bool valid(int remove, int pile)
```

What TYPE of function is this?
bool
Does it RETURN anything? If so, what is the significance of the returned value?
Yes, it returns either true or false, determines if the number of rocks indicated is valid for the pile.
How many PARAMETERS does this function accept?
2
List any parameters that are PASSED BY VALUE to this function:

- remove

- pile

List any parameters that are PASSED BY REFERENCE to this function:

-

turn

```
/*A single turn is played.

For human user, they are prompted to ask how many rocks to remove, and the pile is
adjusted accordingly

For computer,computer makes its decision, the number of rocks it removes is displayed,
and the pile is adjusted accordingly

This function calls helper functions:
-valid
-adjustPile
-compDecision

Parameters:
player – which player's turn it currently is – 1=user 2=computer
pLose – has the human player has lost
cLose – has the computer has lost
pile – total number of rocks remaining in the pile
person - the human player's name
*/
void turn(int player, bool &pLose, bool &cLose, int &pile, string person)
```

What TYPE of function is this?

void

Does it RETURN anything? If so, what is the significance of the returned value?

No

How many PARAMETERS does this function accept?

5

List any parameters that are PASSED BY VALUE to this function:

- player
- person

List any parameters that are PASSED BY REFERENCE to this function:

- pLose
- cLose
- pile

gameOver

```
/*determines whether or not game is over
Outputs message about who has lost, when appropriate
Returns true if the game IS over, false if it is still ongoing
Parameters:
pLose – has the player lost?
cLose – has the computer lost?
*/
bool gameOver(bool pLose, bool cLose)
```

What TYPE of function is this?

bool

Does it RETURN anything? If so, what is the significance of the returned value?

Yes, it returns either true or false, if the game is over or not(t/f).

How many PARAMETERS does this function accept?

2

List any parameters that are PASSED BY VALUE to this function:

- pLose
- cLose

List any parameters that are PASSED BY REFERENCE to this function:

-