

Function name	#	Test description	Sample input			Expected result			Actual result			Pass / fail
			Type	Name	Value / value at address	Type	Name	Value / value at address	Type	Parameter name	Value / value at address	
promptSelectMode	1	nInput < 0	int	nInput	scanf() => -1	int	Invalid input for nInput	Invalid input for nInput	int	nInput	0	Pass
		nInput = 0	int	nInput	scanf() => 0	int	nInput	0	int	nInput	0	Pass
		nInput = 1	int	nInput	scanf() => 1	int	nInput	1	int	nInput	1	Pass
		nInput > 1	int	nInput	scanf() => 2	int	Invalid input for nInput	Invalid input for nInput	int	nInput	1	Pass
promptSetStats	2	Scanned input: 100/10/10/10/15	int *	nTargetHP	scanf() => 100	int *	nTargetHP	100	int *	nTargetHP	100	Pass
			int *	nTargetATK	scanf() => 10	int *	nTargetATK	10	int *	nTargetATK	10	Pass
			int *	nTargetDEF	scanf() => 10	int *	nTargetDEF	10	int *	nTargetDEF	10	Pass
			int *	nTargetSPD	scanf() => 10	int *	nTargetSPD	10	int *	nTargetSPD	10	Pass
		Scanned input: 1000/20/20/20/100	int *	nTargetCrit	scanf() => 15	int *	nTargetCrit	15	int *	nTargetCrit	15	Pass
			int *	nTargetHP	scanf() => 999	int *	Invalid input for nTargetHP	Invalid input for nTargetHP	int *	nTargetHP	999	Pass
			int *	nTargetATK	scanf() => 20	int *	nTargetATK	20	int *	nTargetATK	20	Pass
			int *	nTargetDEF	scanf() => 20	int *	nTargetDEF	20	int *	nTargetDEF	20	Pass
		Scanned input: 999/999/999/999/999	int *	nTargetSPD	scanf() => 20	int *	nTargetSPD	20	int *	nTargetSPD	20	Pass
			int *	nTargetCrit	scanf() => 100	int *	nTargetCrit	100	int *	nTargetCrit	100	Pass
			int *	nTargetHP	scanf() => 999	int *	nTargetHP	999	int *	nTargetHP	999	Pass
			int *	nTargetATK	scanf() => 999	int *	nTargetATK	999	int *	nTargetATK	999	Pass
			int *	nTargetDEF	scanf() => 999	int *	nTargetDEF	999	int *	nTargetDEF	999	Pass
			int *	nTargetSPD	scanf() => 999	int *	nTargetSPD	999	int *	nTargetSPD	999	Pass
			int *	nTargetCrit	scanf() => 999	int *	Invalid input for nTargetHP	Invalid input for nTargetHP	int *	nTargetHP	999	Pass
equipWeapon Preconditions: nInput is 0, 1, 2, or 3	3	nInput = 0	int *	nTargetATK	10	int *	nTargetATK	25	int *	nTargetATK	25	Pass
			int *	nTargetSPD	10	int *	nTargetSPD	13	int *	nTargetSPD	13	Pass
			int *	nInput	0	char *	function return	"Dagger"	char *	function return	"Dagger"	Pass
		nInput = 1	int *	nTargetATK	10	int *	nTargetATK	30	int *	nTargetATK	30	Pass
			int *	nTargetSPD	10	int *	nTargetSPD	10	int *	nTargetSPD	10	Pass
			int *	nInput	1	char *	function return	"Katana"	char *	function return	"Katana"	Pass
		nInput = 2	int *	nTargetATK	10	int *	nTargetATK	50	int *	nTargetATK	50	Pass
			int *	nTargetSPD	10	int *	nTargetSPD	5	int *	nTargetSPD	5	Pass
			int *	nInput	2	char *	function return	"Broadsword"	char *	function return	"Broadsword"	Pass
		nInput = 3	int *	nTargetATK	10	int *	nTargetATK	10	int *	nTargetATK	10	Pass
			int *	nTargetSPD	10	int *	nTargetSPD	20	int *	nTargetSPD	20	Pass
			int *	nInput	3	char *	function return	"Fist"	char *	function return	"Fist"	Pass
equipArmor Preconditions: nInput is 0, 1, 2, or 3	4	nInput = 0	int *	nTargetHP	100	int *	nTargetHP	100	int *	nTargetHP	100	Pass
			int *	nTargetDEF	10	int *	nTargetDEF	13	int *	nTargetDEF	13	Pass
			int *	nTargetSPD	10	int *	nTargetSPD	13	int *	nTargetSPD	13	Pass
		nInput = 1	int *	nInput	0	char *	function return	"Mythril"	char *	function return	"Mythril"	Pass
			int *	nTargetHP	100	int *	nTargetHP	105	int *	nTargetHP	105	Pass
			int *	nTargetDEF	10	int *	nTargetDEF	18	int *	nTargetDEF	18	Pass
		nInput = 2	int *	nTargetSPD	10	int *	nTargetSPD	10	int *	nTargetSPD	10	Pass
			int *	nInput	1	char *	function return	"Chainmail"	char *	function return	"Chainmail"	Pass
			int *	nTargetHP	100	int *	nTargetHP	110	int *	nTargetHP	110	Pass
		nInput = 3	int *	nTargetDEF	10	int *	nTargetDEF	28	int *	nTargetDEF	28	Pass
			int *	nTargetSPD	10	int *	nTargetSPD	6	int *	nTargetSPD	6	Pass
			int *	nInput	2	char *	function return	"Adamantite armor"	char *	function return	"Adamantite armor"	Pass
promptWeaponSelect	5	nInput < 0	int *	nInput	scanf() => -1	int *	Invalid input for nInput	Invalid input for nInput	int *	nInput	0	Pass
			int *	nInput	scanf() => 0	int *	function return	0	int *	function return	0	Pass
			int *	nInput	scanf() => 1	int *	function return	1	int *	function return	1	Pass
		nInput = 2	int *	nInput	scanf() => 2	int *	function return	2	int *	function return	2	Pass
			int *	nInput	scanf() => 3	int *	function return	3	int *	function return	3	Pass
			int *	nInput	scanf() => 4	int *	Invalid input for nInput	Invalid input for nInput	int *	nInput	3	Pass
		nInput > 3	int *	nInput	scanf() => -1	int *	Invalid input for nInput	Invalid input for nInput	int *	nInput	3	Pass
			int *	nInput	scanf() => 0	int *	function return	0	int *	function return	0	Pass
			int *	nInput	scanf() => 1	int *	function return	1	int *	function return	1	Pass
		nInput = 1	int *	nInput	scanf() => 2	int *	function return	2	int *	function return	2	Pass
			int *	nInput	scanf() => 3	int *	function return	3	int *	function return	3	Pass
			int *	nInput	scanf() => 4	int *	Invalid input for nInput	Invalid input for nInput	int *	nInput	3	Pass
promptEquipConfirm	7	cInput = 'Y'	char *	cInput	scanf() => 'Y'	int *	function return	1	int *	function return	1	Pass
			char *	cInput	scanf() => 'y'	int *	function return	1	int *	function return	1	Pass
			char *	cInput	scanf() => 'N'	int *	function return	0	int *	function return	0	Pass
		cInput = 'N'	char *	cInput	scanf() => 'n'	int *	function return	0	int *	function return	0	Pass
			char *	cInput	scanf() => 'A'	int *	Invalid input for cInput	Invalid input for cInput	int *	cInput	0	Pass
			char *	cInput	scanf() => 'A'	int *	Invalid input for cInput	Invalid input for cInput	int *	cInput	0	Pass
		nWeaponInput = 0 AND nArmorInput = 0	int *	nEnemyHP	100	int *	nEnemyHP	100	int *	nEnemyHP	100	Pass
			int *	nEnemyATK	10	int *	nEnemyATK	25	int *	nEnemyATK	25	Pass
			int *	nEnemyDEF	10	int *	nEnemyDEF	13	int *	nEnemyDEF	13	Pass
		nWeaponInput = 1 AND nArmorInput = 1	int *	nEnemySPD	10	int *	nEnemySPD	16	int *	nEnemySPD	16	Pass
			int *	nEnemyCrit	15	int *	nEnemyCrit	15	int *	nEnemyCrit	15	Pass
			int *	nWeaponInput	rand() % 4 => 0	int *	nArmorInput	rand() % 4 => 0	int *	nWeaponInput	15	Pass
promptEnemyEquipConfirm	8	nWeaponInput = 2 AND nArmorInput = 3	int *	nEnemyHP	100	int *	nEnemyHP	100	int *	nEnemyHP	100	Pass
			int *	nEnemyATK	10	int *	nEnemyATK	50	int *	nEnemyATK	50	Pass
			int *	nEnemyDEF	10	int *	nEnemyDEF	10	int *	nEnemyDEF	10	Pass
		nWeaponInput = 3 AND nArmorInput = 3	int *	nEnemySPD	10	int *	nEnemySPD	15	int *	nEnemySPD	15	Pass
			int *	nEnemyCrit	15	int *	nEnemyCrit	15	int *	nEnemyCrit	15	Pass
			int *	nWeaponInput	rand() % 4 => 1	int *	nArmorInput	rand() % 4 => 1	int *	nWeaponInput	15	Pass
		nWeaponInput = 0 AND nArmorInput = 0	int *	nEnemyHP	100	int *	nEnemyHP	100	int *	nEnemyHP	100	Pass
			int *	nEnemyATK	10	int *	nEnemyATK	50	int *	nEnemyATK	50	Pass
			int *	nEnemyDEF	10	int *	nEnemyDEF	10	int *	nEnemyDEF	10	Pass
		nWeaponInput = 1 AND nArmorInput = 1	int *	nEnemySPD	10	int *	nEnemySPD	15	int *	nEnemySPD	15	Pass
			int *	nEnemyCrit	15	int *	nEnemyCrit	15	int *	nEnemyCrit	15	Pass
			int *	nWeaponInput	rand() % 4 => 2	int *	nArmorInput	rand() % 4 => 3	int *	nWeaponInput	15	Pass
processAttack Preconditions: nActorATK and nTargetDEF are non-negative and ≤ 999 nActorCrit is non-negative and ≤ 100	9	nCritRoll ≥ nActorCrit AND nTargetDEF ≤ nActorATK AND nActorATK - nTargetDEF ≤ nTargetCurrentHP	int *	nActorATK	50	int *	nTargetCurrentHI	60	int *	nTargetCurrentHI	60	Pass
			int *	nActorCrit	15	int *	nTargetCurrentHI	100	int *	nTargetCurrentHI	100	Pass
			int *	nTargetDEF	100	int *	nTargetCurrentHI	100	int *	nTargetCurrentHI	100	Pass
		nCritRoll ≥ nActorCrit AND nTargetDEF > nActorATK AND nActorATK - nTargetDEF ≤ nTargetCurrentHP	int *	nActorCrit	15	int *	nTargetCurrentHI	100	int *	nTargetCurrentHI	100	Pass
			int *	nTargetDEF	100	int *	nTargetCurrentHI	100	int *	nTargetCurrentHI	100	Pass
			int *	nActorATK	100	int *	nTargetCurrentHI	100	int *	nTargetCurrentHI	100	Pass
		nCritRoll < nActorCrit AND nActorATK ≤ nTargetCurrentHP	int *	nActorATK	50	int *	nTargetCurrentHI	50	int *	nTargetCurrentHI	50	Pass
			int *	nActorCrit	15	int *	nTargetCurrentHI	50	int *	nTargetCurrentHI	50	Pass
			int *	nTargetDEF	100	int *	nTargetCurrentHI	50	int *	nTargetCurrentHI	50	Pass
		nCritRoll ≥ nActorCrit AND nActorATK - nTargetDEF > nTargetCurrentHP	int *	nActorATK	50	int *	nTargetCurrentHI	50	int *	nTargetCurrentHI	50	Pass
			int *	nActorCrit	15	int *	nTargetCurrentHI	50	int *	nTargetCurrentHI	50	Pass
			int *	nTargetDEF	100	int *	nTargetCurrentHI	50	int *	nTargetCurrentHI	50	Pass
processBlock Preconditions: nActorDEF is non-negative and ≤ 999	10	nActorDEF = 10 AND nActorDEF = 50 AND nActorDEF = 999	int *	nActorDEF	10	int *	nActorDEF	20	int *	nActorDEF	20	Pass
			int *	nActorDEF	50	int *	nActorDEF	100	int *	nActorDEF	100	Pass
			int *	nActorDEF	999	int *	nActorDEF	1998	int *	nActorDEF	1998	Pass
		nActorIsCharging = 0	int *	nActorATK	10	int *	nActorATK	20	int *	nActorATK	20	Pass
			int *	nActorSPD	10	int *	nActorSPD	20	int *	nActorSPD	20	Pass
			int *	nActorIsCharging	0	int *	nActorIsCharging	0	int *	nActorIsCharging	0	Pass
		nActorIsCharging = 1	int *	nActorATK	20	int *	nActorATK	20	int *	nActorATK	20	Pass
			int *	nActorSPD	20	int *	nActorSPD	20	int *	nActorSPD	20	Pass
			int *	nActorIsCharging	1	int *	nActorIsCharging	0	int *	nActorIsCharging	0	Pass
		nActorIsCharging = 0	int *	nActorATK	20	int *	nActorATK	20	int *	nActorATK	20	Pass
			int *	nActorSPD	20	int *	nActorSPD	20	int *	nActorSPD	20	Pass
			int *	nActorIsCharging	0	int *	nActorIsCharging	0	int *	nActorIsCharging	0	Pass
		nActorIsCharging = 1	int *	nActorATK	20	int *	nActorATK	20	int *	nActorATK	20	Pass
			int *	nActorSPD	20	int *	nActorSPD	20	int *	nActorSPD	20	Pass
			int *	nActorIsCharging	1	int *	nActorIsCharging	0	int *	nActorIsCharging	0	Pass

negative and ≤ 999		int	nActorIsPlayer	1					
	nActorATK = nActorSPD =	int *	nActorATK	999	int *	nActorATK	1998	int *	nActorATK
	999	int *	nActorSPD	999	int *	nActorSPD	1998	int *	nActorSPD
	AND	int *	nActorIsCharging	0	int *	nActorIsCharging	0	int *	nActorIsCharging
	nActorIsCharging = 0	int	nActorIsPlayer	1					