				Samo	le input		Evnecte	ed result		Actual res	ule	
Function name	#	Test description	Туре	Name	Value / value at address	Туре	Name	Value / value at address	Туре	Parameter name V	alue / value at address	Pass / fail
promptColoctModo		nInput < 0 nInput = 0	int	ninput	scanf() => -1 scanf() => 0	int	input for ninput ninput	0	int int	nput for ninput ninput	0	Pass Pass
promptSelectMode	1	nInput = 1	int	nInput	scanf() => 1	int	nInput	1	int	nInput	1	Pass
promptSetStats		nInput > 1	int *	nInput nTargetHP	scanf() => 2 scanf() => 100	int *	input for ninput nTargetHP	100	int *	nTargetHP	100	Pass Pass
		Scanned input: 100/10/10/15	int *	nTargetATK	scanf() => 10	int *	nTargetATK	10	int *	nTargetATK	10	Pass
	2		int * int *	nTargetDEF nTargetSPD	scanf() => 10 scanf() => 10	int * int *	nTargetDEF nTargetSPD	10 10	int * int *	nTargetDEF nTargetSPD	10 10	Pass Pass
			int *	nTargetCrit	scanf() => 15	int *	nTargetCrit	15	int *	nTargetCrit	15	Pass
		Scanned input: 1000/20/20/20/100	int *	nTargetHP nTargetATK	scanf() => 999 scanf() => 20		input for nTargetH nTargetATK	IP 20	Invalid in int *	nput for nTargetHP nTargetATK	20	Pass Pass
			int *	nTargetDEF	scanf() => 20	int *	nTargetDEF	20	int *	nTargetDEF	20	Pass
			int *	nTargetSPD	scanf() => 20		nTargetSPD	20 100	int * int *	nTargetSPD	20 100	Pass
		Scanned input: 999/999/999	int *	nTargetCrit nTargetHP	scanf() => 100 scanf() => 999	int *	nTargetCrit nTargetHP	999	int *	nTargetCrit nTargetHP	999	Pass Pass
			int *	nTargetATK	scanf() => 999	int *	nTargetATK	999	int *	nTargetATK	999	Pass
			int *	nTargetDEF nTargetSPD	scanf() => 999 scanf() => 999	int * int *	nTargetDEF nTargetSPD	999 999	int * int *	nTargetDEF nTargetSPD	999 999	Pass Pass
			int *	nTargetCrit	scanf() => 999	Invalid	input for nTargetH			put for nTargetHP		Pass
	3	nInput = 0	int *	nTargetATK nTargetSPD	10 10		nTargetATK nTargetSPD	25 13	int * int *	nTargetATK nTargetSPD	25 13	Pass Pass
			int	nInput	0	char *	function return	"Dagger"	char *	function return	"Dagger"	Pass
		nInput = 1	int * int *	nTargetATK nTargetSPD	10 10	int * int *	nTargetATK nTargetSPD	30 10	int * int *	nTargetATK nTargetSPD	30 10	Pass Pass
equipWeapon Preconditions: nInput is 0, 1, 2, or 3			int	ninput	1	char *	function return	"Katana"	char *	function return	"Katana"	Pass
	-	nInput = 2	int * int *	nTargetATK nTargetSPD	10 10	int *	nTargetATK nTargetSPD	50 5	int * int *	nTargetATK nTargetSPD	50 5	Pass Pass
		minput = 2	int	ninput	2		function return	"Broadsword"	char *	function return	"Broadsword"	Pass
		alassa 2	int *	nTargetATK	10	int *	nTargetATK	10	int *	nTargetATK	10	Pass
		nInput = 3	int *	nTargetSPD nInput	10		nTargetSPD function return	20 "Fist"	int * char *	nTargetSPD function return	20 "Fist"	Pass Pass
equinarmor		ninput = 0	int *	nTargetHP	100	int *	nTargetHP	100	int *	nTargetHP	100	Pass
			int *	nTargetDEF nTargetSPD	10 10	int * int *	nTargetDEF nTargetSPD	13 13	int * int *	nTargetDEF	13 13	Pass Pass
			int	nInput	0	char *	function return	"Mythril"	char *	nTargetSPD function return	"Mythril"	Pass
		-	int *	nTargetHP	100	int *	nTargetHP	105	int *	nTargetHP	105	Pass
		nInput = 1	int * int *	nTargetDEF nTargetSPD	10 10		nTargetDEF nTargetSPD	18 10	int * int *	nTargetDEF nTargetSPD	18 10	Pass Pass
equipArmor Preconditions:	4		int	nInput	1	char *	function return	"Chainmail"	char *	function return	"Chainmail"	Pass
nInput is 0, 1, 2, or 3			int * int *	nTargetHP nTargetDEF	100 10	int * int *	nTargetHP nTargetDEF	110 28	int * int *	nTargetHP nTargetDEF	110 28	Pass Pass
		nInput = 2	int *	nTargetSPD	10	int *	nTargetSPD	6	int *	nTargetSPD	6	Pass
			int *	nInput nTargetHP	2 100		function return nTargetHP	"Adamantite armor"	char *	function return ". nTargetHP	Adamantite armor" 100	Pass Pass
		nInput = 3	int *	nTargetDEF	10		nTargetDEF	2	int *	nTargetDEF	2	Pass
		minput = 3	int *	nTargetSPD	10		nTargetSPD	20 "No armor"	int * char *	nTargetSPD	20 "No armor"	Pass
promptWeaponSelect		nInput < 0	int	ninput ninput	scanf() => -1		function return	NO armor		function return put for ninput	NO armor	Pass Pass
		ninput = 0	int	nInput	scanf() => 0	int	function return	0	int	function return	0	Pass
	5	nInput = 1 nInput = 2	int	ninput	scanf() => 1 scanf() => 2	int int	function return function return	1 2	int int	function return function return	1 2	Pass
		nInput = 3	int	ninput	scanf() => 3	int	function return	3	int	function return	3	Pass
		ninput > 3	int int	nInput	scanf() => 4		input for nInput			put for ninput		Pass Pass
promptArmorSelect		nInput < 0 nInput = 0	int	ninput	scanf() => -1 scanf() => 0	int	function return	0	int	function return	0	Pass
	6	ninput = 1	int	nInput	scanf() => 1	int	function return	1	int	function return	1	Pass
		nInput = 2 nInput = 3	int	ninput	scanf() => 2 scanf() => 3	int	function return function return	2	int	function return function return	3	Pass Pass
		nInput > 3	int	nInput	scanf() => 4		input for nInput			put for nInput		Pass
		cinput = 'Y'	char :	cinput cinput	scanf() => 'Y' scanf() => 'y'	int	function return function return	11	int	function return function return	1 1	Pass Pass
promptEquipConfirm	7	cinput = 'N'	char :	cinput	scanf() => 'N'	int	function return	0	int	function return	0	Pass
		cinput = 'n' cinput = 'A'	char :	cinput cinput	scanf() => 'n' scanf() => 'A'	int	function return input for clnput	0	int Invalid in	function return put for cinput	0	Pass
		p	int *	nEnemyHP	100	int *	nEnemyHP	100	int *	nEnemyHP	100	Pass
		nWeaponInput = 0	int *	nEnemyATK nEnemyDEF	10 10	int * int *	nEnemyATK nEnemyDEF	25 13	int * int *	nEnemyATK nEnemyDEF	25 13	Pass Pass
		AND	int *	nEnemySPD	10	int *	nEnemySPD	16	int *	nEnemySPD	16	Pass
		nArmorInput = 0	int *	nEnemyCrit	15	int *	nEnemyCrit	15	int *	nEnemyCrit	15	Pass
			int int	nWeaponInput nArmorInput	rand() % 4 => 0 rand() % 4 => 0							
			int *	nEnemyHP	100	int *	nEnemyHP	105	int *	nEnemyHP	105	Pass
promptEnemyEquipConfirm		nWeaponInput = 1	int * int *	nEnemyATK nEnemyDEF	10 10	int *	nEnemyATK nEnemyDEF	30 18	int * int *	nEnemyATK nEnemyDEF	30 18	Pass Pass
	8	AND	int *	nEnemySPD	10	int *	nEnemySPD	10	int *	nEnemySPD	10	Pass
		nArmorInput = 1	int *	nEnemyCrit nWeaponInput	15 rand() % 4 => 1	int *	nEnemyCrit	15	int *	nEnemyCrit	15	Pass
			int	nArmorInput	rand() % 4 => 1							
			int *	nEnemyHP nEnemyATK	100 10		nEnemyHP nEnemyATK	100 50	int * int *	nEnemyHP nEnemyATK	100 50	Pass Pass
		nWeaponInput = 2	int *	nEnemyDEF	10	int *	nEnemyDEF	10	int *	nEnemyDEF	10	Pass
		AND nArmorInput = 3	int * int *	nEnemySPD nEnemyCrit	10 15		nEnemySPD nEnemyCrit	15 15	int * int *	nEnemySPD nEnemyCrit	15 15	Pass Pass
			int	nEnemyCrit nWeaponInput	rand() % 4 => 2	IIIL *	nemyent	15	IIIC .	nemycht	15	r d55
		nCritDoll > nA storCrit	int	nArmorInput	rand() % 4 => 3	det *	aTargat*		int *	wTorgotC		D
		nCritRoll ≥ nActorCrit AND	int int	nActorATK nActorCrit	50 15	TUC 4	nTargetCurrentH	11 60	int *	nTargetCurrentHI	60	Pass
		nTargetDEF ≤ nActorATK	int *	nTargetCurrentHP	100							
		AND nActorATK - nTargetDEF ≤	int int	nTargetDEF nActorIsPlayer	10 1							
		nTargetCurrentHP	int	nCritRoll	rand() % 3 => 0							
		nCritRoll ≥ nActorCrit AND	int int	nActorATK nActorCrit	50 15	int *	nTargetCurrentH	100	int *	nTargetCurrentHI	100	Pass
		nTargetDEF > nActorATK	int *	nTargetCurrentHP	100							
processAttack		AND nActorATK - nTargetDEF ≤	int int	nTargetDEF nActorIsPlayer	60 1							
Preconditions:		nTargetCurrentHP	int	nCritRoll	rand() % 3 => 1							
	9	margetearrentm		nActorATK	50	int *	nTargetCurrentH	11 50	int *	nTargetCurrentHI	50	Pass
nActorATK and nTargetDEF are non- negative and ≤ 999	9		int	nActorCrit	4.5							
nActorATK and nTargetDEF are non- negative and ≤ 999	9	nCritRoll < nActorCrit	int int int *	nActorCrit nTargetCurrentHP	15 100							
nActorATK and nTargetDEF are non- negative and ≤ 999	9	nCritRoll < nActorCrit AND nActorATK ≤	int int * int	nTargetCurrentHP nTargetDEF	100 10							
nActorATK and nTargetDEF are non- negative and ≤ 999	9	nCritRoll < nActorCrit AND	int int * int int	nTargetCurrentHP nTargetDEF nActorIsPlayer	100 10 1							
nActorATK and nTargetDEF are non- negative and ≤ 999	9	nCritRoll < nActorCrit AND nActorATK ≤ nTargetCurrentHP	int * int * int int int int int	nTargetCurrentHP nTargetDEF nActorIsPlayer nCritRoll nActorATK	100 10 1 rand() % 3 => 2 999	int *	nTargetCurrentH	ii 0	int *	nTargetCurrentHI	0	Pass
nActorATK and nTargetDEF are non- negative and ≤ 999	9	nCritRoll < nActorCrit AND nActorATK ≤	int * int int int int int int	nTargetCurrentHP nTargetDEF nActorIsPlayer nCritRoll nActorATK nActorCrit	100 10 1 rand() % 3 => 2 999 15	int *	nTargetCurrentH	ii 0	int *	nTargetCurrentHI	0	Pass
nActorATK and nTargetDEF are non- negative and ≤ 999	9	nCritRoll < nActorCrit AND nActorATK ≤ nTargetCurrentHP nCritRoll ≥ nActorCrit	int * int * int int int int int	nTargetCurrentHP nTargetDEF nActorIsPlayer nCritRoll nActorATK	100 10 1 rand() % 3 => 2 999	int *	nTargetCurrentH	II 0	int *	nTargetCurrentHI	0	Pass
nActorATK and nTargetDEF are non- negative and ≤ 999	9	nCritRoll < nActorCrit AND nActorATK ≤ nTargetCurrentHP nCritRoll ≥ nActorCrit AND	int * int	nTargetCurrentHP nTargetDEF nActorIsPlayer nCritRoll nActorATK nActorCrit nTargetCurrentHP nTargetDEF nActorIsPlayer	100 10 1 rand() % 3 => 2 999 15 100 10	int *	nTargetCurrentH	II 0	int *	nTargetCurrentHI	0	Pass
nActorATK and nTargetDEF are non- negative and ≤ 999	9	nCritRoll < nActorCrit AND nActorATK ≤ nTargetCurrentHP nCritRoll ≥ nActorCrit AND nActorATK - nTargetDEF >	int * int	nTargetCurrentHP nTargetDEF nActorIsPlayer nCritRoll nActorATK nActorCrit nTargetCurrentHP nTargetDEF	100 10 1 1 rand() % 3 => 2 999 15 100 10		nTargetCurrentH	9	int *	nTargetCurrentHI nActorDEF	0 20	Pass
nActorATk and nTargetDEF are non- negative and ≤ 999 nActorCrit is non-negative and ≤ 100 processBlock processBlock Preconditions:	9	nCritRoll < nActorCrit AND nActorATK ≤ nTargetCurrentHP nCritRoll ≥ nActorCrit AND nActorATK - nTargetDEF > nTargetCurrentHP nActorDEF = 10 nActorDEF = 50	int int * int	nTargetCurrentHP nTargetDEF nActorIsPlayer nCritRoll nActorATK nActorCrit nTargetCurrentHP nTargetDEF nActorIsPlayer nCritRoll nActorDEF nActorDEF	100 10 11 1 rand() % 3 => 2 999 15 100 10 1 rand() % 3 => 3 10 50	int *	nActorDEF nActorDEF	20 100	int *	nActorDEF nActorDEF	20 100	Pass Pass
nActorXT and nTargetDEF are non- negative and ≤ 999 nActorCrit is non-negative and ≤ 100 processBlock processBlock Preconditions:		nCritRoll < nActorCrit AND nActorATK ≤ nTargetCurrentHP nCritRoll ≥ nActorCrit AND nActorATK - nTargetDEF > nTargetCurrentHP	int int * int	nTargetCurrentHP nTargetDEF nActorIsPlayer nCritRoll nActorATK nActorCrit nTargetCurrentHP nTargetDEF nActorIsPlayer nCritRoll nActorDEF nActorDEF nActorDEF	100 10 11 rand() % 3 => 2 999 15 10 10 11 rand() % 3 => 3 10 10 50 999	int * int * int *	nActorDEF nActorDEF nActorDEF	20 100 1998	int * int * int *	nActorDEF nActorDEF nActorDEF	20 100 1998	Pass Pass Pass
nActorXT and nTargetDF are non- negative and ≤ 99 nActorCrit is non-negative and ≤ 100 processBlock processBlock Preconditions:		nCritRoll < nActorCrit AND nActorATK ≤ nTargetCurrentHP nCritRoll ≥ nActorCrit AND nActorATK - nTargetDEF > nTargetCurrentHP nActorDEF = 10 nActorDEF = 50 nActorDEF = 999	int	nTargetCurrentHP nTargetDEF nActorisPlayer nCritRoll nActorATK nActorCritRoll nTargetCurrentHP nTargetDEF nActorCritRoll nActorCritRoll nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF	100 10 11 rand() % 3 *> 2 999 15 100 10 11 rand() % 3 *>> 3 10 50 999 10 10	int * int * int * int * int *	nActorDEF nActorDEF nActorDEF nActorATK nActorSPD	20 100 1998 20 20	int * int * int * int * int *	nActorDEF nActorDEF nActorDEF nActorATK nActorSPD	20 100 1998 20 20	Pass Pass Pass Pass Pass Pass
nActorXX and nTargetDEF are non- negative and ≤ 99 nActorCrit is non-negative and ≤ 100 processBlock processBlock Preconditions:		nCritRoll < nActorCrit AND nActorATK ≤ nTargetCurrentHP nCritRoll ≥ nActorCrit AND nActorATK - nTargetDEF > nTargetCurrentHP nActorDEF = 10 nActorDEF = 50	int	nTargetDEF nActorsPlayer nCritRoll nActorsPlayer nCritRoll nActorCrit nTargetDEF nActorSPlayer nActorDEF nActorDEF nActorDEF nActorDEF nActorTK nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF nActorSPlayen nActorSPlayen nActorSPlayen nActorSPlayen nActorSPlayen nActorSPlayen nActorSPlayen nActorSPlayen nActorSPlayen nActorScharging	100 100 11 rand() % 3 => 2 999 15 100 10 11 rand() % 3 => 3 10 999 10 10 10 10 10 10 10 10 10 10 10 10 10	int * int * int * int * int *	nActorDEF nActorDEF nActorDEF nActorATK	20 100 1998 20 20	int * int * int *	nActorDEF nActorDEF nActorDEF nActorATK	20 100 1998 20	Pass Pass Pass Pass
nActorXT and nTargetDF are non- negative and ≤ 99 nActorCrit is non-negative and ≤ 100 processBlock processBlock Preconditions:		nCritRoll < nActorCrit AND nActorATK ≤ nTargetCurrentHP nCritRoll ≥ nActorCrit AND nActorATK - nTargetDEF > nTargetCurrentHP nActorDEF = 10 nActorDEF = 50 nActorDEF = 999	int int * int	nTargetCurrentHP nTargetDEF nActorisPlayer nCritRoll nActorATK nActorCritRoll nTargetCurrentHP nTargetDEF nActorCritRoll nActorCritRoll nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF	100 10 11 rand() % 3 *> 2 999 15 100 10 11 rand() % 3 *>> 3 10 50 999 10 10	int *	nActorDEF nActorDEF nActorDEF nActorATK nActorSPD nActorIsCharging	20 100 1998 20 20	int * int * int * int * int *	nActorDEF nActorDEF nActorDEF nActorATK nActorSPD	20 100 1998 20 20	Pass Pass Pass Pass Pass Pass
nActorXx and nTargetDEF are non- negative and ≤ 99 nActorCrit is non-negative and ≤ 100 nActorCrit is non-negative and ≤ 100 processBlock processBlock Preconditions: nActorDEF is non-negative and ≤ 999		nCritRoll < nActorCrit AND nActorATK ≤ nTargetCurrentHP nCritRoll ≥ nActorCrit AND nActorATK - nTargetDEF > nTargetCurrentHP nActorDEF = 10 nActorDEF = 50 nActorDEF = 999	int	nTargetDEF nActorisPlayer nCritRoll nActorisPlayer nCritRoll nActorATK nActorCrit nTargetDEF nActorisPlayer nCritRoll nActorDEF nActorDEF nActorDEF nActorDEF nActorDEF nActorSPD nActorSPD nActorSPD nActorSPD nActorSPD nActorSPlayer	100 10 10 10 10 17 17 17 15 100 10 11 17 10 10 10 10 10 10 10 10 10 10 10 10 10	int *	nActorDEF nActorDEF nActorATK nActorSPD nActorIsCharging	20 100 1998 20 20 3	int * int * int * int * int * int *	nActorDEF nActorDEF nActorDEF nActorATK nActorSPD nActorIsCharging	20 100 1998 20 20 0	Pass Pass Pass Pass Pass Pass Pass

negative and ≤ 999

int nActorIsPlayer
int * nActorATK
int * nActorSPD
int * nActorIsCharging
int nActorIsPlayer nActorATK = nActorSPD = 999 AND nActorIsCharging = 0 1998 int * nActorATK 1998 int * nActorSPD 0 int * nActorIsCharging 1998 1998 0 Pass Pass Pass