

| Function name | | # | Test description | Sample input | | Expected result | | Actual result | | Pass / fail | | | |
|------------------------------|--|---|--------------------------------|--------------|--------------------------|-----------------|--------------------------------|--------------------------|---|-------------|-----------------|--------------------|------|
| | | | Type | Name | Value / value at address | Type | Name | Value / value at address | Parameter name Value / value at address | | | | |
| int promptSelectMode | | 1 | scanf() nInput < 0 | int | nInput | scanf() => -1 | Invalid input for nInput | | Invalid input for nInput | Pass | | | |
| | | | scanf() nInput = 0 | int | nInput | scanf() => 0 | int | function return | 0 | int | function return | 0 | Pass |
| | Input conditions: nInput is from scanf() nInput is 0 or 1 | | scanf() nInput = 1 | int | nInput | scanf() => 1 | int | function return | 1 | int | function return | 1 | Pass |
| | | | scanf() nInput > 1 | int | nInput | scanf() => 2 | Invalid input for nInput | | Invalid input for nInput | | | Pass | |
| | | | scanf() 100/10/10/10/15 | int * | nTargetHP | scanf() => 100 | int * | nTargetHP | 100 | int * | nTargetHP | 100 | Pass |
| void promptSetStats | | 2 | scanf() 1000/20/20/20/100 | int * | nTargetATK | scanf() => 10 | int * | nTargetATK | 10 | int * | nTargetATK | 10 | Pass |
| | | | scanf() 1000/20/20/20/100 | int * | nTargetDEF | scanf() => 10 | int * | nTargetDEF | 10 | int * | nTargetDEF | 10 | Pass |
| | Input conditions: Inputs are from scanf() *nTargetHP, *nTargetATK, *nTargetDEF, and *nTargetSPD are less than or equal to 999 *nTargetCrit is less than or equal to 100 | | scanf() 999/999/999/999/999 | int * | nTargetSPD | scanf() => 10 | int * | nTargetSPD | 10 | int * | nTargetSPD | 10 | Pass |
| | | | scanf() 999/999/999/999/999 | int * | nTargetCrit | scanf() => 15 | int * | nTargetCrit | 15 | int * | nTargetCrit | 15 | Pass |
| | | | scanf() 999/999/999/999/999 | int * | nTargetHP | scanf() => 1000 | Invalid input for *nTargetHP | | Invalid input for *nTargetHP | | | Pass | |
| char * equipWeapon | | 3 | scanf() nInput = 0 | int * | nTargetATK | scanf() => 20 | int * | nTargetATK | 20 | int * | nTargetATK | 20 | Pass |
| | | | scanf() nInput = 0 | int * | nTargetDEF | scanf() => 20 | int * | nTargetDEF | 20 | int * | nTargetDEF | 20 | Pass |
| | Preconditions: nInput is 0, 1, 2, or 3 *nTargetATK and *nTargetSPD are non- negative and less than or equal to 999 | | scanf() nInput = 1 | int * | nTargetSPD | scanf() => 20 | int * | nTargetSPD | 20 | int * | nTargetSPD | 20 | Pass |
| | | | scanf() nInput = 1 | int * | nTargetCrit | scanf() => 100 | int * | nTargetCrit | 100 | int * | nTargetCrit | 100 | Pass |
| | | | scanf() nInput = 1 | int * | nTargetHP | scanf() => 999 | int * | nTargetHP | 999 | int * | nTargetHP | 999 | Pass |
| char * equipArmor | | 4 | scanf() nInput = 0 | int * | nTargetATK | scanf() => 999 | int * | nTargetATK | 999 | int * | nTargetATK | 999 | Pass |
| | | | scanf() nInput = 0 | int * | nTargetDEF | scanf() => 999 | int * | nTargetDEF | 999 | int * | nTargetDEF | 999 | Pass |
| | Preconditions: nInput is 0, 1, 2, or 3 *nTargetHP, *nTargetDEF, and *nTargetSPD are non-negative and less than or equal to 999 | | scanf() nInput = 1 | int * | nTargetSPD | scanf() => 999 | int * | nTargetSPD | 999 | int * | nTargetSPD | 999 | Pass |
| | | | scanf() nInput = 1 | int * | nTargetCrit | scanf() => 999 | Invalid input for *nTargetCrit | | Invalid input for *nTargetCrit | | | Pass | |
| | | | scanf() nInput = 1 | int * | nTargetHP | scanf() => 999 | int * | nTargetHP | 999 | int * | nTargetHP | 999 | Pass |
| int promptWeaponSelect | | 5 | scanf() nInput < 0 | int * | nTargetATK | scanf() => 25 | int * | nTargetATK | 25 | int * | nTargetATK | 25 | Pass |
| | | | scanf() nInput = 0 | int * | nTargetSPD | scanf() => 13 | int * | nTargetSPD | 13 | int * | nTargetSPD | 13 | Pass |
| | Input conditions: nInput is from scanf() nInput is 0, 1, 2, or 3 | | scanf() nInput = 1 | int * | nInput | scanf() => 0 | char * | function return | "Dagger" | char * | function return | "Dagger" | Pass |
| | | | scanf() nInput = 1 | int * | nTargetATK | scanf() => 30 | int * | nTargetATK | 30 | int * | nTargetATK | 30 | Pass |
| | | | scanf() nInput = 1 | int * | nTargetSPD | scanf() => 10 | int * | nTargetSPD | 10 | int * | nTargetSPD | 10 | Pass |
| int promptArmorSelect | | 6 | scanf() nInput = 2 | int * | nTargetHP | scanf() => 1 | char * | function return | "Katana" | char * | function return | "Katana" | Pass |
| | | | scanf() nInput = 2 | int * | nTargetATK | scanf() => 10 | int * | nTargetATK | 10 | int * | nTargetATK | 10 | Pass |
| | Preconditions: nInput is 0, 1, 2, or 3 *nTargetHP, *nTargetDEF, and *nTargetSPD are non-negative and less than or equal to 999 | | scanf() nInput = 2 | int * | nTargetSPD | scanf() => 50 | int * | nTargetSPD | 50 | int * | nTargetSPD | 50 | Pass |
| | | | scanf() nInput = 2 | int * | nTargetCrit | scanf() => 10 | char * | function return | "Broadsword" | char * | function return | "Broadsword" | Pass |
| | | | scanf() nInput = 2 | int * | nTargetHP | scanf() => 10 | int * | nTargetHP | 10 | int * | nTargetHP | 10 | Pass |
| int promptPlayerEquipConfirm | | 7 | scanf() nInput < 0 | int * | nTargetSPD | scanf() => 20 | int * | nTargetSPD | 20 | int * | nTargetSPD | 20 | Pass |
| | | | scanf() nInput = 0 | int * | nInput | scanf() => 3 | char * | function return | "Fist" | char * | function return | "Fist" | Pass |
| | Input conditions: cInput is from scanf() cInput is 'Y', 'y', 'N', or 'n' | | scanf() cInput = 1 | int * | nTargetATK | scanf() => 100 | int * | nTargetATK | 100 | int * | nTargetATK | 100 | Pass |
| | | | scanf() cInput = 1 | int * | nTargetDEF | scanf() => 10 | int * | nTargetDEF | 10 | int * | nTargetDEF | 10 | Pass |
| | | | scanf() cInput = 1 | int * | nTargetSPD | scanf() => 10 | int * | nTargetSPD | 10 | int * | nTargetSPD | 10 | Pass |
| void promptEnemyEquipConfirm | | 8 | scanf() cInput = 1 | int * | nInput | scanf() => 1 | char * | function return | "Chainmail" | char * | function return | "Chainmail" | Pass |
| | | | scanf() cInput = 1 | int * | nTargetHP | scanf() => 110 | int * | nTargetHP | 110 | int * | nTargetHP | 110 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 2 | int * | nTargetDEF | scanf() => 28 | int * | nTargetDEF | 28 | int * | nTargetDEF | 28 | Pass |
| | | | scanf() cInput = 2 | int * | nTargetSPD | scanf() => 6 | int * | nTargetSPD | 6 | int * | nTargetSPD | 6 | Pass |
| | | | scanf() cInput = 2 | int * | nInput | scanf() => 2 | char * | function return | "Adamantite armor" | char * | function return | "Adamantite armor" | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nTargetHP | scanf() => 100 | int * | nTargetHP | 100 | int * | nTargetHP | 100 | Pass |
| | | | scanf() cInput = 3 | int * | nTargetDEF | scanf() => 10 | int * | nTargetDEF | 10 | int * | nTargetDEF | 10 | Pass |
| | Input conditions: nWeaponInput and nArmorInput are from rand() % 4 nWeaponInput and nArmorInput are 0, 1, 2, or 3 | | scanf() cInput = 3 | int * | nTargetSPD | scanf() => 10 | int * | nTargetSPD | 10 | int * | nTargetSPD | 10 | Pass |
| | | | scanf() cInput = 3 | int * | nTargetCrit | scanf() => 15 | int * | nTargetCrit | 15 | int * | nTargetCrit | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| void processAttack | | 9 | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | Pass |
| | | | scanf() cInput = 3 | int * | nArmorInput | scanf() => 15 | int * | nArmorInput | 15 | int * | nArmorInput | 15 | Pass |
| | Preconditions: *nEnemyHP, *nEnemyATK, *nEnemyDEF, and *nEnemySPD are non-negative and less than or equal to 999 | | scanf() cInput = 3 | int * | nWeaponInput | scanf() => 15 | int * | nWeaponInput | 15 | int * | nWeaponInput | 15 | |

| | | | | | | | | | | |
|--|----|--|---|--|----------------------------|---|----------------------|---|----------------------|-----------------------------|
| <pre> nActorRoll is non-negative and less than or equal to 100 nActorIsPlayer is 0 or 1 Input conditions: nCritRoll is from rand() % 100 nCritRoll ranges from 0 to 99 </pre> | | | <pre> nActorATK ≤ nTargetCurrentHP </pre> | <pre> int nActorATK int nActorCrit int * nTargetCurrentHP </pre> | <pre> 50 15 100 10 </pre> | <pre> int * nTargetCurrentHI </pre> | <pre> 50 </pre> | <pre> int * nTargetCurrentHI </pre> | <pre> 50 </pre> | <pre> Pass </pre> |
| | | | <pre> rand() % 100 nCritRoll < nActorCrit </pre> | <pre> int nActorIsPlayer nCritRoll </pre> | <pre> 1 1 </pre> | | | | | |
| | | | <pre> nActorATK - nTargetDEF > nTargetCurrentHP </pre> | <pre> int nActorATK int nActorCrit int * nTargetCurrentHP </pre> | <pre> 999 15 100 10 </pre> | <pre> int * nTargetCurrentHI </pre> | <pre> 0 </pre> | <pre> int * nTargetCurrentHI </pre> | <pre> 0 </pre> | <pre> Pass </pre> |
| | | | <pre> rand() % 100 nCritRoll ≥ nActorCrit </pre> | <pre> int nActorIsPlayer nCritRoll </pre> | <pre> 1 1 </pre> | | | | | |
| | 10 | void processBlock | | | | | | | | |
| | | Preconditions: | | | | | | | | |
| | | *nActorDEF is non-negative and less than or equal to 999 | | | | | | | | |
| | | nActorIsPlayer is 0 or 1 | | | | | | | | |
| | | | nActorDEF = 10 | int * nActorDEF | 10 | int * nActorDEF | 20 | int * nActorDEF | 20 | Pass |
| | | | nActorDEF = 50 | int * nActorDEF | 50 | int * nActorDEF | 100 | int * nActorDEF | 100 | Pass |
| | | | nActorDEF = 999 | int * nActorDEF | 999 | int * nActorDEF | 1998 | int * nActorDEF | 1998 | Pass |
| | | | | | | | | | | |
| <pre> nActorATK and nTargetSPD are non- negative and ≤ 999 </pre> | | | <pre> nActorATK = 10 nActorSPD = 10 nActorIsCharging = 0 </pre> | <pre> int * nActorATK int * nActorSPD int * nActorIsCharging int nActorIsPlayer </pre> | <pre> 10 10 0 A </pre> | <pre> int * nActorATK int * nActorSPD int * nActorIsCharging </pre> | <pre> 20 20 0 </pre> | <pre> int * nActorATK int * nActorSPD int * nActorIsCharging </pre> | <pre> 20 20 0 </pre> | <pre> Pass Pass Pass </pre> |
| | | void processCharge | | | | | | | | |
| | | Preconditions: | | | | | | | | |
| | | nActorATK and nTargetSPD are non-negative and ≤ 999 | | | | | | | | |
| | 11 | | nActorIsCharging = 1 | int * nActorATK int * nActorSPD int * nActorIsCharging int nActorIsPlayer | 20 20 1 1 | int * nActorATK int * nActorSPD int * nActorIsCharging | 20 20 1 | int * nActorATK int * nActorSPD int * nActorIsCharging | 20 20 0 | Pass Pass Pass |
| | | | nActorATK = 999 | int * nActorATK | 999 | int * nActorATK | 1998 | int * nActorATK | 1998 | Pass |
| | | | nActorSPD = 999 | int * nActorSPD | 999 | int * nActorSPD | 1998 | int * nActorSPD | 1998 | Pass |
| | | | nActorIsCharging = 0 | int * nActorIsCharging int nActorIsPlayer | 0 1 | int * nActorIsCharging | 0 | int * nActorIsCharging | 0 | Pass |

Result

Program progresses

Program enters **Standard mode**

Broadsword is selected for player

Program progresses

```
[=====]
|                                     |
|               Select an armor.     |
|                                     |
|=====|=====|
| Armor | Details |
|-----|-----|
|[0] Mythril | Lightweight armor that increases DEF and SPD. |
|            | (DEF + 3, SPD + 3) |
|-----|-----|
|[1] Chainmail | Standard chainmail armor that boosts DEF and HP |
|            | (DEF + 8, HP + 5) |
|-----|-----|
|[2] Adamantite armor | Heavy armor than boosts DEF and HP; lowers SPD. |
|            | (DEF + 18, HP + 10, SPD - 4) |
|-----|-----|
|[3] No armor | Wearing no armor greatly improves speed. |
|            | (SPD + 10) |
|=====|=====|

|=====|=====|
| Choose. (0/1/2/3) | Current stats: 100/10/10/10 |
|=====|=====|
>>
```

User enters 3 No armor is selected for player

```
[=====]
|                                     |
|               You selected the following armor: |
|               No armor |
|=====|
>> Press Enter to proceed.
```

User presses enter key Program progresses

```
[=====]
|               You will battle with the following stats: |
|=====|

|=====|=====|=====|
| Stats | Weapon | Armor |
|-----|-----|-----|
| ATK \ DEF \ SPD | Broadsword | No armor |
| 50 / 10 / 15 | | |
|-----|-----|-----|
| Base HP: 100 | | |
| Critical chance: 15% | | |
|=====|=====|=====|

|=====|=====|=====|
| Proceed? (Y/y/N/n) | TIP: Canceling will let you change equipment. |
|=====|=====|=====|
>>
```

User enters Y Program progresses

```
[=====]
|                                     |
|               The enemy is selecting their equipment! |
|                                     |
|=====|
>> Press Enter to proceed.
```

User presses enter key Program progresses

```
[=====]
|               The enemy has chosen the following equipment: |
|=====|

|=====|=====|=====|
| Stats | Weapon | Armor |
|-----|-----|-----|
| ATK \ DEF \ SPD | Dagger | Mythril |
| 25 / 13 / 16 | | |
|-----|-----|-----|
| Base HP: 100 | | |
| Critical chance: 15% | | |
|=====|=====|=====|

>> Press Enter to proceed.
```

Program generates value for **rand() % 4** **Dagger** and **Mythril** are selected for enemy

User presses enter key Program progresses

```
[=====]
|                                     |
|               The enemy gladiator challenges you! |
|               Reduce their HP to 0 to win! |
|=====|
>> Press Enter to proceed.
```

User presses enter key Program progresses

```

=====
Enter an action (0/1/2).
=====
[=====]
[ You VS Enemy ]
[ ##### ]
[ HP: 100 / 100 ]
[ ##### ]
[ ATK \ DEF \ SPD ]
[ 50 / 10 / 15 ]
[ Critical chance: 15% ]
[ ##### ]
[ ##### ]
[ 0 0 ]
[ /\ /\ ]
[ /\ /\ ]
[ ##### ]
[ [0] Attack | Deal damage to the enemy. ]
[ [1] Block | Brace yourself. (Doubles DEF this turn) ]
[ [2] Charge | Charge power. (Doubles ATK and SPD next turn) ]
[ ##### ]
>>

```

User enters 2

Charge action is selected for player

Program generates value for
rand() % 3

Block action is selected for enemy

```

=====
Enemy blocked! DEF doubled this turn.
=====
[=====]
[ You VS Enemy ]
[ ##### ]
[ HP: 100 / 100 ]
[ ##### ]
[ ATK \ DEF \ SPD ]
[ 50 / 10 / 15 ]
[ Critical chance: 15% ]
[ ##### ]
[ ##### ]
[ 0 0 ]
[ /\ /\ ]
[ /\ /\ ]
[ ##### ]
[ ##### ]
[ >> Press Enter to proceed. ]
[ ##### ]

```

Enemy executes **Block** action

Enemy **DEF** is doubled

User presses enter key

Program progresses

```

=====
You charged! ATK and SPD doubled until the end of the next turn.
=====
[=====]
[ You VS Enemy ]
[ ##### ]
[ HP: 100 / 100 ]
[ ##### ]
[ ATK \ DEF \ SPD ]
[ 100 / 10 / 30 ]
[ Critical chance: 15% ]
[ ##### ]
[ ##### ]
[ 0 0 ]
[ /\ /\ ]
[ /\ /\ ]
[ ##### ]
[ ##### ]
[ >> Press Enter to proceed. ]
[ ##### ]

```

Player executes **Charge** action

Player **ATK** and **SPD** are doubled

User presses enter key

Program progresses

```

=====
Enter an action (0/1/2).
=====
[=====]
[ You VS Enemy ]
[ ##### ]
[ HP: 100 / 100 ]
[ ##### ]

```

Turn begins

Enemy **DEF** is reverted

Player **charging flag** is enabled

```
[=====  
| ATK  \ DEF  \ SPD |  
| 100  / 10   / 30 |  
| Critical chance: 15% |  
|=====|  
  
|      0      |  
|    /\    |  
|   /\   |  
  
|=====|  
| [0] Attack | Deal damage to the enemy.  
| [1] Block  | Brace yourself. (Doubles DEF this turn)  
| [2] Charge | Charge power. (Doubles ATK and SPD next turn)  
|=====|  
>>
```

User enters 0

Attack action is selected for player

Program generates value for **rand() % 3**

Attack action is selected for enemy

```
[=====  
| You attacked! Dealt 87 damage. |  
|=====|  
|      You      |      VS      |      Enemy      |  
| ##### |  
| HP: 100 / 100 |  
|-----|  
| ATK  \ DEF  \ SPD |  
| 100  / 10   / 30 |  
| Critical chance: 15% |  
|=====|  
  
|      0      |  
|    /\    |  
|   /\   |  
  
|=====|  
|=====|  
|=====|  
>> Press Enter to proceed.
```

Player executes **Attack** action

Enemy **HP** is reduced by
Player **ATK** - Enemy **DEF**
= 100 - 13
= 87

User presses enter key

Program progresses

```
[=====  
| Enemy attacked! Dealt 15 damage. |  
|=====|  
|      You      |      VS      |      Enemy      |  
| ##### |  
| HP: 85 / 100 |  
|-----|  
| ATK  \ DEF  \ SPD |  
| 100  / 10   / 30 |  
| Critical chance: 15% |  
|=====|  
  
|      0      |  
|    /\    |  
|   /\   |  
  
|=====|  
|=====|  
|=====|  
>> Press Enter to proceed.
```

Enemy executes **Attack** action

Player **HP** is reduced by
Enemy **ATK** - Player **DEF**
= 25 - 10
= 15

User presses enter key

Program progresses

```
[=====  
| Enter an action (0/1/2). |  
|=====|  
|      You      |      VS      |      Enemy      |  
| ##### |  
| HP: 85 / 100 |  
|-----|  
| ATK  \ DEF  \ SPD |  
| 50   / 10   / 15 |  
| Critical chance: 15% |  
|=====|  
  
|      0      |  
|    /\    |  
|   /\   |  
  
|=====|  
|=====|  
|=====|
```

Turn begins

Player **ATK** and **SPD** are reverted

User enters 1

Block action is selected for player

Block action is selected for enemy

$$\begin{array}{c} 0 \\ / \quad | \quad \backslash \\ / \quad \backslash \end{array}$$

Player **DEF** is doubled

```
>> Press Enter to proceed.
```

Program progresses

$$\begin{array}{c} 0 \\ / \quad | \quad \backslash \\ / \quad \backslash \end{array}$$

Enemy **DEF** is doubled

```
>> Press Enter to proceed.
```

Program progresses

$$\begin{array}{c} 0 \\ / \quad | \quad \backslash \\ / \quad \backslash \end{array}$$

Enemy **DEF** is reverted

Attack action is selected for player

>>

Charge action is selected for enemy

