



# JUPAN

—— J U P A N   D U N G E O N   (v3.7) ——

 던전을 통해 주판 읽는법을 배워보자 . 

[ P R E S S   S T A R T ]

© 2025   Geunseok Soft.   All rights reserved.





# 목차

- 기본 주판 표시
- 계산 모드 (덧셈/ 뺄셈)
- 랜덤 숫자 던전

# 시작 화면

# JUPAN

— J U P A N D U N G E O N (v3.7) —

 던전을 통해 주판 읽는법을 배워보자 . 

[ P R E S S   S T A R T ]

© 2025 Geunseok Soft. All rights reserved.

# 기본 주판 표시

입력된 숫자를 주판 모양으로 출력합니다.  
0을 입력하면 메뉴로 돌아갑니다.

숫자 입력> |

입력된 값 : 1234

0	0	0	0	0	1	2	3	4
○	○	○	○	○	○	○	○	○
<hr/>								
○	○	○	○	○	●	●	●	●
○	○	○	○	○	○	●	●	●
○	○	○	○	○	○	○	●	●
○	○	○	○	○	○	○	○	●
○	○	○	○	○	○	○	○	○

(Enter 키를 눌러 계속...)|

# 계산 모드(덧셈/뺄셈)



주판 계산 모드

1. 덧셈 | 2. 뺄셈 | 0. 뒤로가기

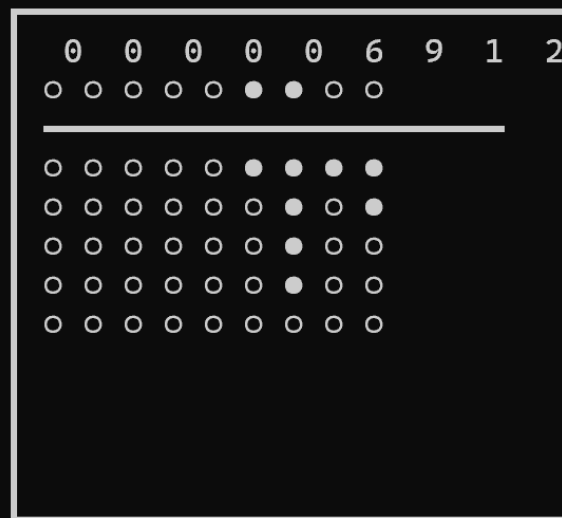
선택 > |

# 계산 모드 (예시)

+ 덧셈 모드입니다 (0 입력 시 종료)

첫 번째 수 입력 > 1234  
두 번째 수 입력 > 5678|

결과 : 1234 + 5678 = 6912



(Enter 키를 눌러 계속...)|

# 던전 모드

주판 던전 - 전투 모드  
점수 : 0 | 연속 정답 : 0

플레이어 HP: 100/100 [#####]

적 HP: 123/123 [#####]



정답 입력 (0 입력 시 종료, 제한시간 10초)> |

# 던전 게임 규칙

## ✂ 주판 던전 전투 규칙 ✂

- 1 주판에 표시된 숫자를 맞히면 적에게 피해를 줍니다.
- 2 틀리거나 제한시간(10초)이 지나면 당신이 피해를 입습니다.
- 3 적 체력이 30% 이하일 때는 제한시간이 5초로 줄어듭니다.
- 4 연속 정답 시 피해량이 2배씩 증가합니다.
- 5 0을 입력하면 전투가 종료됩니다.

Enter 키를 눌러 전투를 시작하세요...



# 결과 값

주판 던전 - 전투 모드  
점수: 0 | 연속 정답: 0

플레이어 HP: 87/100 [#####]

적 HP: 123/123 [#####]



정답 입력 (0 입력 시 종료, 제한시간 10초)> 18765

✅ 정답! 적에게 5 피해! (적 HP: 118)

(Enter 키를 눌러 계속)

주판 던전 - 전투 모드  
점수: 0 | 연속 정답: 0

플레이어 HP: 100/100 [#####]

적 HP: 123/123 [#####]



정답 입력 (0 입력 시 종료, 제한시간 10초)>

🕒 시간 초과! 당신은 13 피해를 입었습니다. (HP: 87)

(Enter 키를 눌러 계속)

# 결과 값

```
주판 던전 - 전투 모드
점수: 5 | 연속 정답: 0

플레이어 HP: 71/100 [#####]
적 HP: 118/123 [#####]

[
  ○ ○ ○ ○ ○ ○ ○ ○ ○
  ───────────
  ○ ○ ○ ○ ● ● ● ● ●
  ○ ○ ○ ○ ○ ○ ● ● ●
  ○ ○ ○ ○ ○ ○ ○ ● ●
  ○ ○ ○ ○ ○ ○ ○ ○ ●
  ○ ○ ○ ○ ○ ○ ○ ○ ○
]

정답 입력 (0 입력 시 종료, 제한시간 10초)> 15516

❌ 오답! 정답은 11234 | 당신은 11 피해를 입었습니다. (HP: 60)

(Enter 키를 눌러 계속)
```

```
===== 전투 결과 =====
최종 점수: 246
최고 연속 정답: 5
=====
(Enter 키를 눌러 메뉴로 돌아가기)|
```

# 코드 분석

주판 숫자 문제 랜덤 생성

제한시간 입력 처리

HP/점수 시스템

콤보 시스템

```

void div_number(int mc[], long number)
{
    for (int i = 0; i <= 8; i++)
    {
        mc[i] = (int)(number / pow(10, 8 - i));
        number -= (long)pow(10, 8 - i) * mc[i];
    }
}

void display_abacus(int m[], int x, int y, int showNumber)
{
    int i, j;
    if (showNumber)
    {
        gotoxy(x, y);
        for (i = 0; i <= 8; i++) printf("%2d ", m[i]);
    }

    gotoxy(x, y + 1);
    for (i = 0; i <= 8; i++)
        if (m[i] / 5) printf("● "); else printf("○ ");

    gotoxy(x, y + 2);
    printf("_____");

    for (i = 0; i <= 8; i++) m[i] %= 5;

    for (i = 1; i <= 5; i++)
    {
        gotoxy(x, y + 2 + i);
        for (j = 0; j <= 8; j++)
            if (m[j] >= i) printf("● "); else printf("○ ");
    }
}

```

## 주판 숫자 문제 주판 생성

- div\_number(int mc[], long number)
- -> 0~9사이의 각 자리 숫자를 왼쪽부터 9칸 채워넣음
- display\_abacus(int m[], int x, int y, int showNumber)
- M의 9자리 숫자를 주판화
- long number를 rand()함수를 활용하여 무작위 숫자화-> 랜덤 주판 완성

# 공격 또는 피해

---

```
if (ans == num)
{
    enemyHP -= (int)dmg;
    score += dmg;
    streak++;
    if (streak > best_streak) best_streak = streak;
    printf("✅ 정답! 적에게 %lld 피해! (적 HP: %d)\n", dmg, enemyHP);
}
else
{
    int p_dmg = 10 + rand() % 11;
    playerHP -= p_dmg;
    streak = 0;
    printf("❌ 오답! 정답은 %ld | 당신은 %d 피해를 입었습니다. (HP: %d)\n", num, p_dmg, playerHP);
}

gotoxy(3, 27);
printf("(Enter 키를 눌러 계속)");
_getch();
}
```

# 제한 시간내 입력처리

- **buf** : 사용자가 입력한 문자열을 저장할 버퍼
- **idx** : 현재 입력된 문자 개수 (버퍼 인덱스)
- **start** : 제한시간 계산용 시작 시각
- **expired** : 시간 초과 여부 플래그

```
idx = 0;
memset(buf, 0, sizeof(buf));

time_t start = time(NULL);
ans = 0;
int expired = 0;
```

# 제한 시간내 입력 처리

---

- time(NULL) : 현재 시각을 초 단위로 반환
- difftime() : 현재시간 - 시작시간 계산 → 경과 시간
- 만약 경과 시간이 제한시간 이상이면 expired = 1로 표시하고 루프 탈출

```
// 제한시간 내 입력 처리
while (1)
{
    double elapsed = difftime(time(NULL), start);
    if (elapsed >= time_limit)
    {
        expired = 1;
        break;
    }

    if (_kbhit())
    {
        int ch = _getch();
        if (ch == 13) break; // Enter
        else if (ch == 8 && idx > 0)
        {
            idx--;
            printf("\b \b");
        }
        else if (ch >= '0' && ch <= '9' && idx < (int)sizeof(buf) - 1)
        {
            buf[idx++] = (char)ch;
            putchar(ch);
        }
    }

    Sleep(100);
}
```

# 업그레이드 사항

---

단일 main함수->모듈형  
구조로 변환(4개의 모드)

숫자입력후 주판변환-  
>학습+게임 통합형 프로그램

입력값만 반복처리-  
>memset,labs,fflush를 통해  
예외 루프처리 등 입력  
버퍼/시간 관리 보안



# 이후 개선점

---

하드 모드 개발( 적 npc 체력 증가 or  
제한 시간 감소(10->7초))

---

남아 있는 시간 화면에 표시

---

보스 몬스터 개발

---

데미지증가 or 힌트를 보여주는 등  
아이템 상점 개발 및 화폐 추가

# 출처 및 참고사항

- <https://m.blog.naver.com/zelojoa96/221354100848https://m.blog.naver.com/zelojoa96/221354100848>  
-참고사항