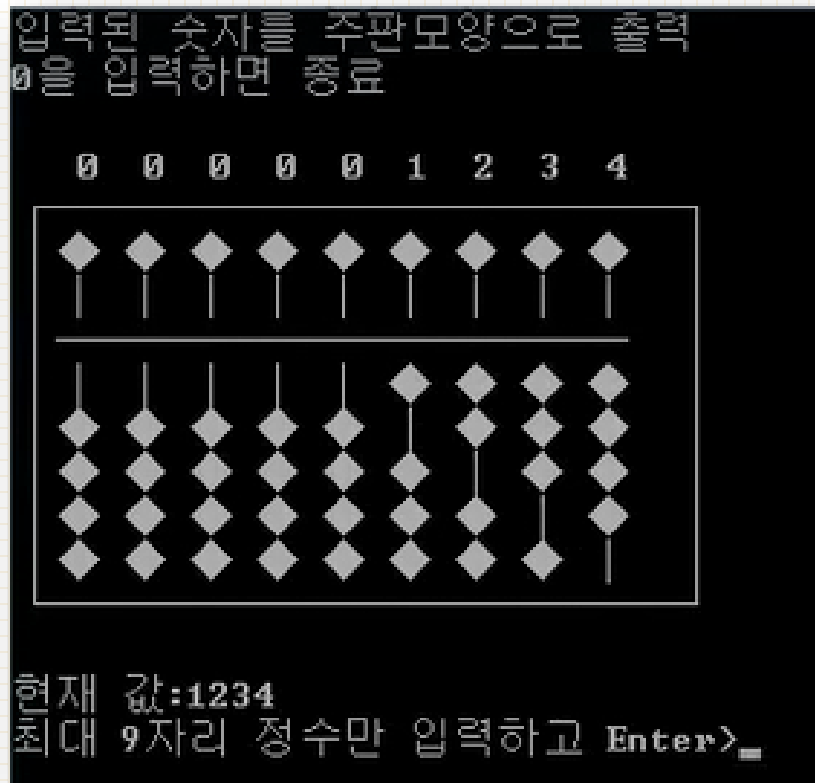


게임 프로그래밍 응용11_1_1.cpp

게임프로그래밍 - 2023864041 정지민

응용11_1_1.cpp를 활용한 주판 프로그램

< Before >

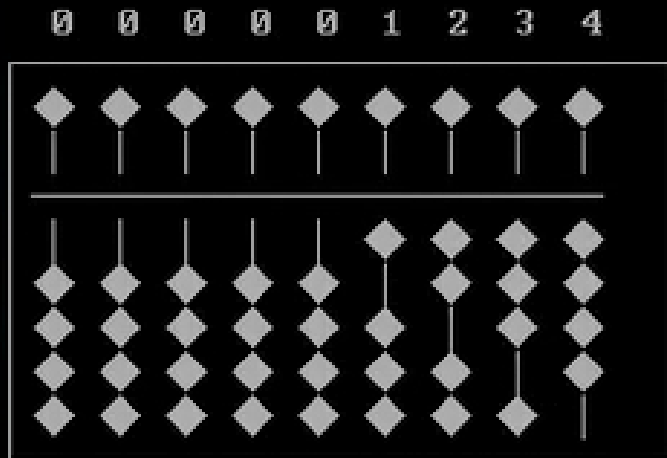


< After >



기본 소스 코드 분석 (응용11_1_1.cpp)

입력된 숫자를 주판모양으로 출력
0을 입력하면 종료



현재 값:1234
최대 9자리 정수만 입력하고 Enter>_

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <conio.h>
#include <windows.h>

void draw_rectangle(int c, int r);
void display_abacus(int m[]);
void div_number(int mc[], long number);
void gotoxy(int x, int y);

int main(void)
{
    long number=0;
    int mc[9];
    printf("입력된 숫자를 주판모양으로 출력\n");
    printf("0을 입력하면 종료\n");
    div_number(mc, number);
    gotoxy(1, 5);
    draw_rectangle(14, 8);
    display_abacus(mc);
    do
    {
        gotoxy(1, 16);
        printf("현재 값:%ld\n", number);
        gotoxy(1, 17);
        printf("최대 9자리 정수만 입력하고 Enter> ");
        gotoxy(34, 17);
        scanf("%ld", &number);
        if (number==0)
            break;
        div_number(mc, number);
        display_abacus(mc);
    }while(1<=number);
    gotoxy(1,17);
    return 0;
}
```

기본 소스 코드 분석 (응용11_1_1.cpp)

```
void div_number(int mc[], long number)
{
    //[함수 11.1.1]의 정의 부분 참고    // number 숫자를 자릿수 별로 구분해
    int i;                             // mc[] 배열에 저장
    for(i=0;i<=8;i++)
    {
        mc[i]= (int) (number/pow(10, 8-i));
        number= number-(long) pow(10, 8-i)*mc[i];
    }
}
```

```
void draw_rectangle(int c, int r)
{
    //[함수 3.5.2]의 정의 부분 참고    // c, r 크기로 직사각형 테두리 출력
    int i, j;
    unsigned char a=0xa6;
    unsigned char b[7];
    for(i=1;i<7;i++)
        b[i]=0xa0+i;

    printf("%c%c",a, b[3]);
    for(i=0;i<c;i++)
        printf("%c%c", a, b[1]);
    printf("%c%c", a, b[4]);
    printf("\n");
    for(i=0;i<r;i++)
    {
        printf("%c%c", a, b[2]);
        for(j=0;j<c;j++)
            printf(" ");
        printf("%c%c",a, b[2]);
        printf("\n");
    }
    printf("%c%", a, b[6]);
    for(i=0;i<c;i++)
        printf("%c%", a, b[1]);
    printf("%c%", a, b[5]);
    printf("\n");
}
```

```
void display_abacus(int m[])
{
    //[함수 11.1.2]의 정의 부분 참고    // 자릿수로 구분된 숫자 배열을 받아
    int i, j;                             // 주판 형태로 출력
    unsigned char a1=0xa6, a2=0xa1;
    gotoxy(3, 4);
    for(i=0;i<=8;i++)
        printf("%2d ", m[i]);
    gotoxy(3, 6);
    for(i=0;i<=8;i++)
        if (m[i]/5)
            printf("%c%c ",a1, 0xa2);
        else
            printf("%c%c ", a2, 0xdf);
    gotoxy(3, 7);
    for(i=0;i<=8;i++)
        if (m[i]<5)
            printf("%c%c ",a1, 0xa2);
        else
            printf("%c%c ", a2, 0xdf);
    for(i=0;i<=8;i++)
        m[i]*=5;
    gotoxy(3, 8);
    for(i=1;i<=13;i++)
        printf("%c%c",a1, a2);
    for(i=1;i<=5;i++)
    {
        gotoxy(3, 8+i);
        for(j=0;j<=8;j++)
            if (m[j]>=i)
                printf("%c%c ", a2, 0xdf);
            else if (m[j]==0) {
                printf("%c%c ", a1, 0xa2);
                m[j]=5;
            }
            else {
                printf("%c%c ", a1, 0xa2);
                m[j]=5;
            }
    }
}
```

기본 소스 코드 분석 (응용11_1_1.cpp)

```
void display_abacus(int m[])
{
    //[함수 11.1.2]의 정의 부분 참고
    int i, j;
    unsigned char a1=0xa6, a2=0xa1;
    gotoxy(3, 4);
    for(i=0;i<=8;i++)
        printf("%2d ", m[i]);
    gotoxy(3, 6);
    for(i=0;i<=8;i++)
        if (m[i]/5)
            printf("%c%c ", a1, 0xa2);
        else
            printf("%c%c ", a2, 0xdf);
}
```

코드	EUC-KR	유니코드	출력 문자	용도
0xA6A1	KS X 1001 (33-33)	U+250C	┌	좌상 모서리, 구분선
0xA6A2	(33-34)	U+2500	—	수평선 또는 막대
0xA1DF	(02-63)	U+25A0	■	구슬(알) 표현용
0xA6A3	(33-35)	U+2510	┐	우상 모서리
0xA6A4	(33-36)	U+2502		세로선
0xA6A5	(33-37)	U+2514	└	좌하 모서리
0xA6A6	(33-38)	U+2518	┘	우하 모서리

소스 코드 수정 1 - 문자 출력 형식 변경

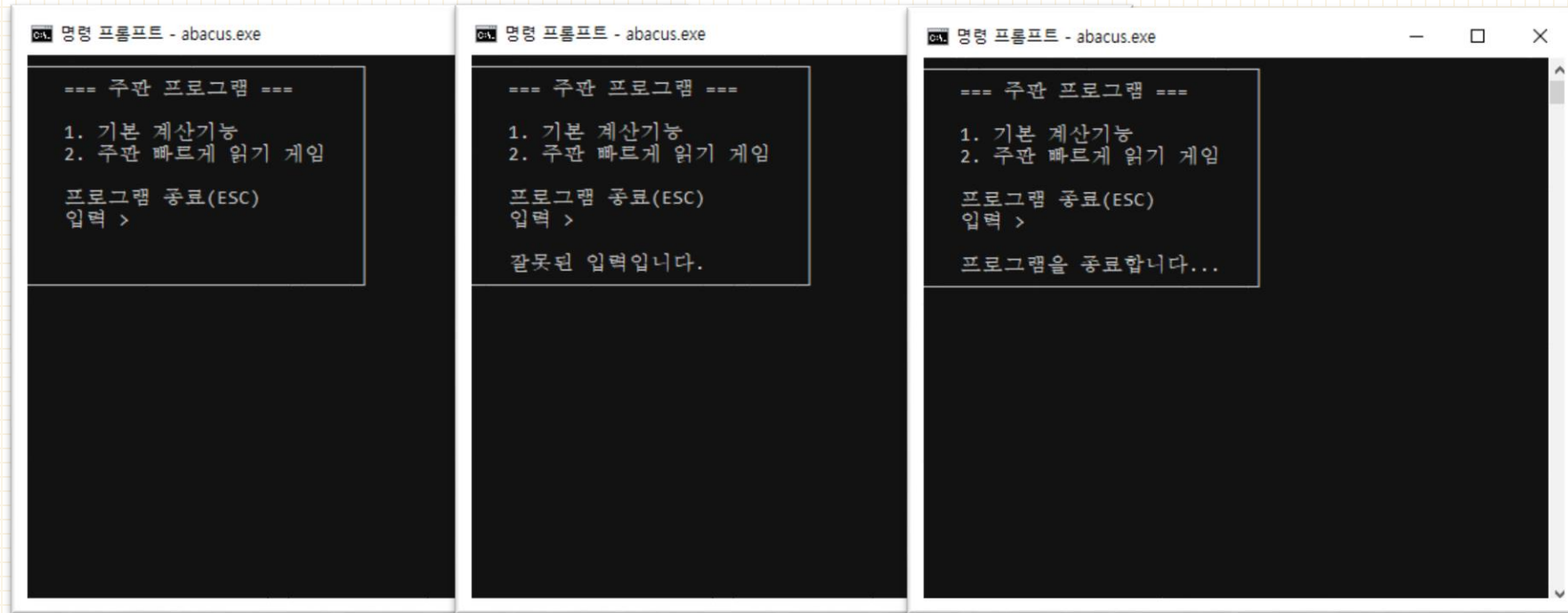
```
tx 0x 0000000000000000
:0
0x 90 00 tx Enter>
```

! EUC-KR 기반의 2바이트 코드 !

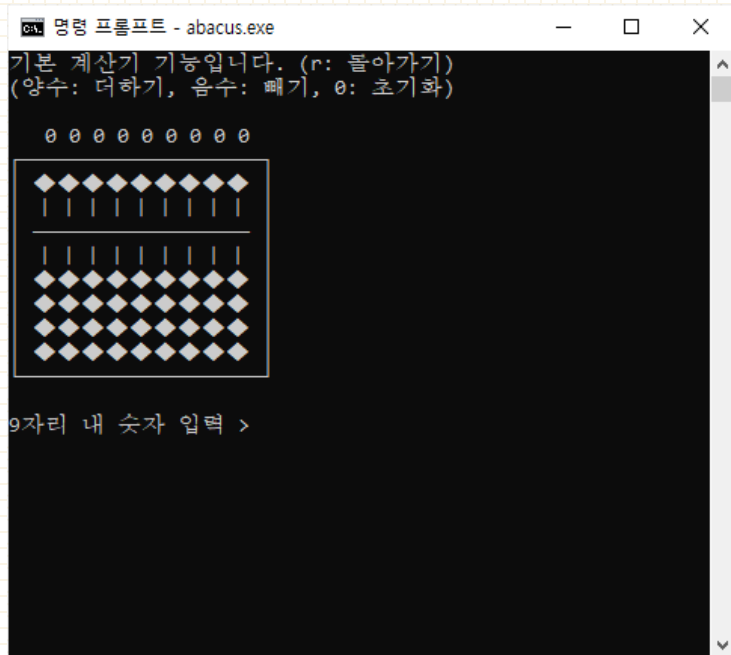
UTF-8 환경에서 문자 깨짐 현상 발생.

-> " | ", "◆", 등의 유니코드로 수정

소스 코드 수정 2 - 메인화면 추가



소스 코드 수정 3 - 계산기 기능



```
// ===== 기본 계산기 기능 =====
void basic_calculator()
{
    long number = 0, InputNumber = 0;
    int mc[9];
    system("cls");
    div_number(mc, number);
    gotoxy(1, 5);
    draw_rectangle(20, 8);
    gotoxy(1, 4);
    printf(" ");
    for (int i = 0; i <= 8; i++)
        printf("%2d", mc[i]);
    display_abacus(mc);

    gotoxy(1, 1);
    printf("기본 계산기 기능입니다. (r: 돌아가기)\n");
}
```

```
while (1)
{
    gotoxy(1, 2);
    printf("(양수: 더하기, 음수: 빼기, 0: 초기화)\n");
    gotoxy(1, 16);
    printf(" ");
    gotoxy(1, 16);
    printf("9자리 내 숫자 입력 >");
    fflush(stdin);

    char buf[20];
    fgets(buf, sizeof(buf), stdin); // 문자열로 입력 받음

    if (buf[0] == 'r' || buf[0] == 'R') // r 또는 R 입력 시 돌아가기
        break;

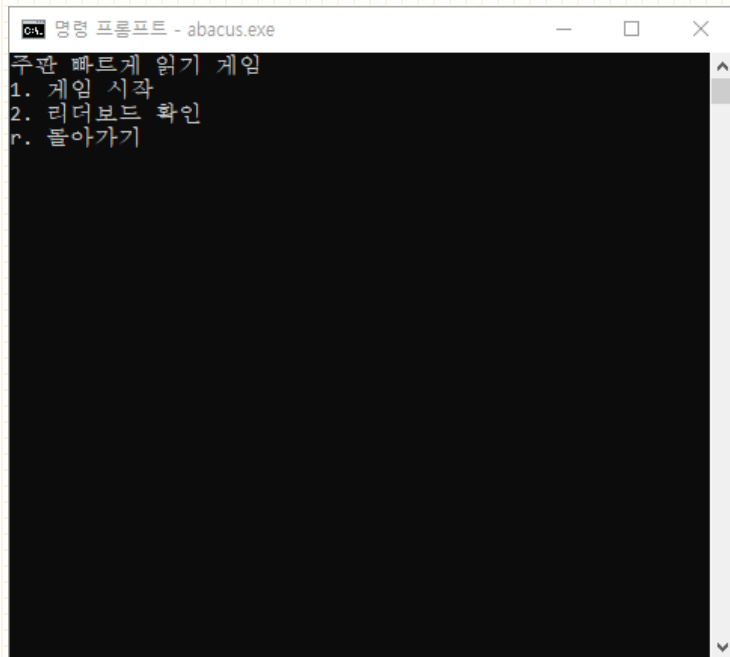
    long InputNumber = strtol(buf, NULL, 10); // 문자열 -> 숫자

    if (InputNumber == 0) number = 0; // 초기화
    else number += InputNumber;

    if (number > 999999999 || number < 0)
    {
        gotoxy(1, 18); printf("계산 범위를 초과했습니다.\n");
        Sleep(1000);
        number -= InputNumber; // 되돌리기
        gotoxy(1, 18); printf(" ");
        continue;
    }

    div_number(mc, number);
    gotoxy(1, 4);
    printf(" ");
    for (int i = 0; i <= 8; i++)
        printf("%2d", mc[i]);
    display_abacus(mc);
}
```

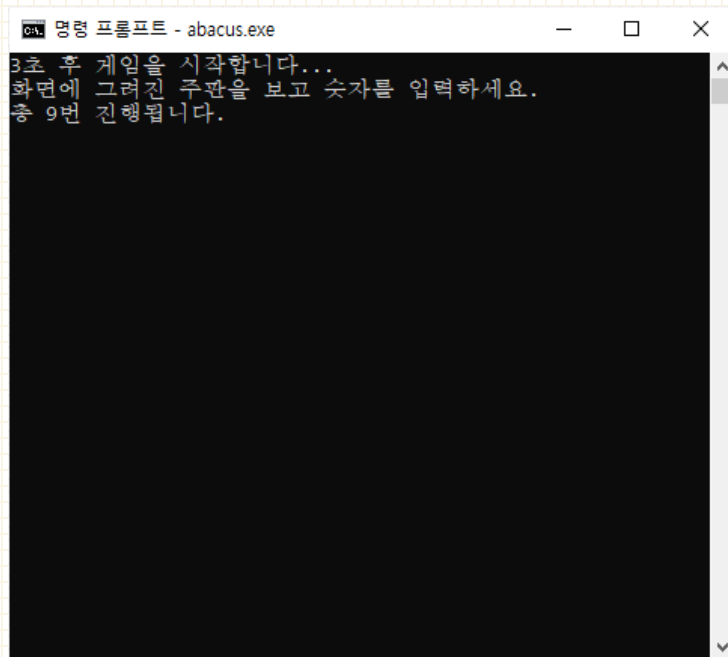

소스 코드 수정 4 - 주판 빠르게 읽기 게임 (0. 게임 메뉴)



```
// ===== 주판 빠르게 읽기 게임 기능 =====
void abacus_speed_game()
{
    int choice;
    system("cls");
    gotoxy(1,1);
    printf("주판 빠르게 읽기 게임\n");
    gotoxy(1,2);
    printf("1. 게임 시작\n2. 리더보드 확인\nr. 돌아가기\n");

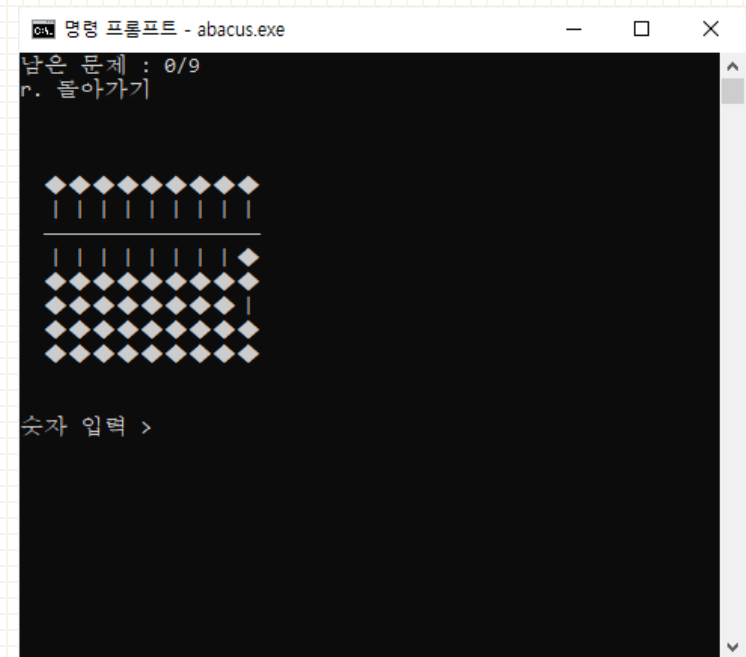
    while (1)
    {
        choice = _getch();
        if (choice == '1') break;      // 게임 시작
        else if (choice == '2')      // 리더보드 확인
        {
            system("cls");
            display_leaderboard("leaderboard.txt");
            printf("아무 키나 누르면 돌아갑니다.\n");
            _getch();
            system("cls");
            gotoxy(1,1);
            printf("주판 빠르게 읽기 게임\n");
            gotoxy(1,2);
            printf("1. 게임 시작\n2. 리더보드 확인\nr. 돌아가기\n");
        }
        else if (choice == 'r' || choice == 'R') return; // 돌아가기
    }
}
```

소스 코드 수정 5 - 주판 빠르게 읽기 게임 (1-1. 게임 시작)

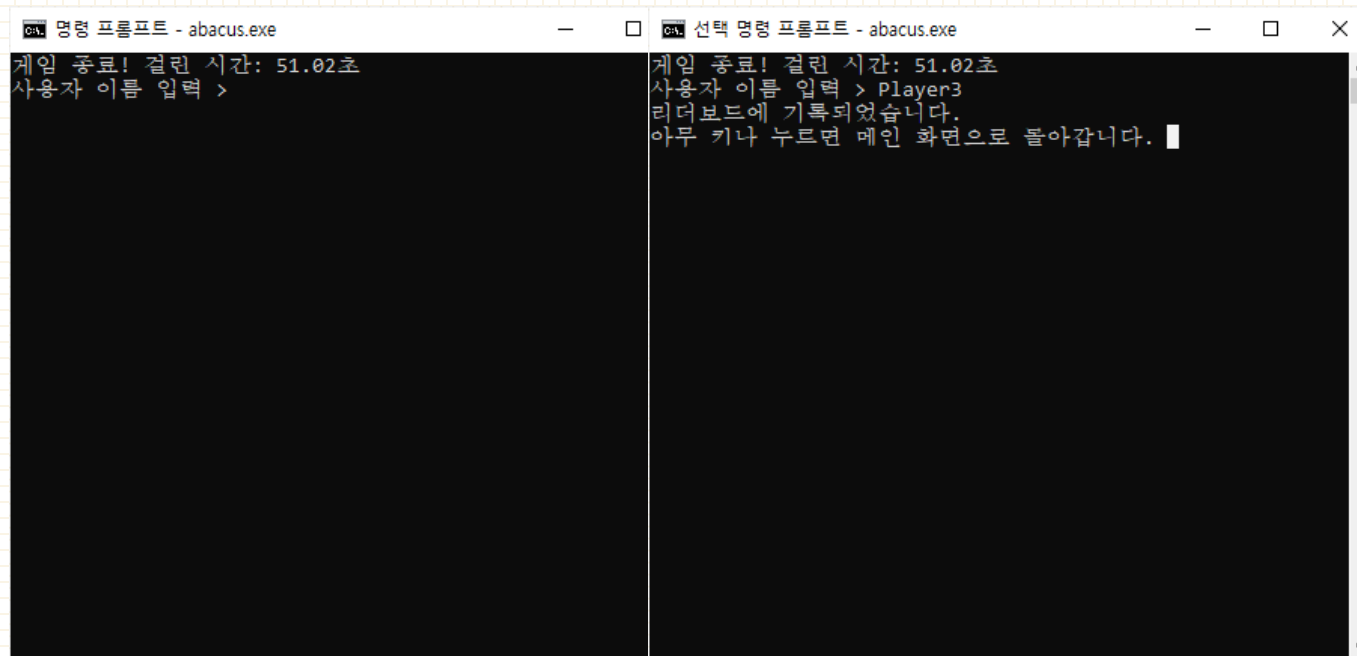


3초 카운트 다운 출력 후

->

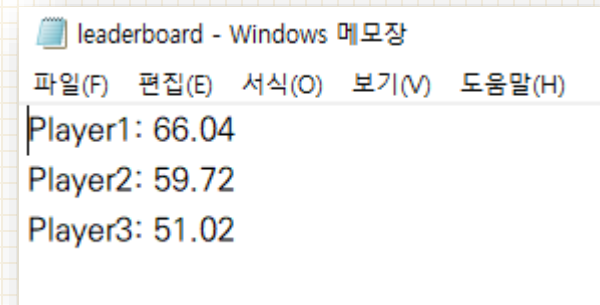


소스 코드 수정 5 - 주판 빠르게 읽기 게임 (1-2. 게임 종료 후 리더보드 저장)



```
명령 프롬프트 - abacus.exe
게임 종료! 걸린 시간: 51.02초
사용자 이름 입력 >

선택 명령 프롬프트 - abacus.exe
게임 종료! 걸린 시간: 51.02초
사용자 이름 입력 > Player3
리더보드에 기록되었습니다.
아무 키나 누르면 메인 화면으로 돌아갑니다.
```



"leaderboard.txt"

소스 코드 수정 6 - 주판 빠르게 읽기 게임 (2. 리더보드 확인)

명령 프롬프트 - abacus.exe

리더보드 (시간 적은 순)

1. Player3: 51.02초
2. Player2: 59.72초
3. Player1: 66.04초

아무 키나 누르면 돌아갑니다.

```
void save_leaderboard(const char* filename, const char* username, double elapsed)
{
    FILE *fp = fopen(filename, "a");
    if (fp)
    {
        fprintf(fp, "%s: %.2f\n", username, elapsed);
        fclose(fp);
    }
}
```

```
int compare_records(const void *a, const void *b)
{
    double t1 = ((Record*)a)→time;
    double t2 = ((Record*)b)→time;
    if (t1 < t2) return -1;
    else if (t1 > t2) return 1;
    else return 0;
}
```

```
void display_leaderboard(const char* filename)
{
    Record records[100];
    int count = 0;

    FILE *fp = fopen(filename, "r");
    if (fp)
    {
        while (fscanf(fp, "%49[^\n]: %lf\n", records[count].name, &records[count].time) == 2)
        {
            count++;
        }
        fclose(fp);

        // 시간 기준 오름차순 정렬
        qsort(records, count, sizeof(Record), compare_records);

        printf("리더보드 (시간 적은 순)\n");
        for (int i = 0; i < count; i++)
        {
            printf("%2d. %s: %.2f초\n", i + 1, records[i].name, records[i].time);
        }
    }
    else
    {
        printf("리더보드가 없습니다.\n");
    }
}

int compare_records(const void *a, const void *b)
{
    double t1 = ((Record*)a)→time;
    double t2 = ((Record*)b)→time;
    if (t1 < t2) return -1;
    else if (t1 > t2) return 1;
    else return 0;
}
```

프로그램 시연

프로그램 시연

화면 전환

새로운 화면 출력 시 기존 화면 깔끔하게 정돈하기

Sleep을 적극 활용하여 알림 메시지 출력하기

어느 세부 상황에서든 프로그램을 종료할 수 있는 방안 마련 (프로그램 종료 ESC, 뒤로가기 R)