

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void ReadInput(void);
void ConvertToArray(void);
void CalculateFinalGrade(void);
void SortByLastName(void);
void PrintArray(char *);
void SortByFinalScore(void);

typedef struct StudentData
{
    struct StudentData *Next;
    char *LastName;
    char *FirstName;
    char *SocialSecurity;
    int Grade1;
    int Grade2;
    int Grade3;
    int Total;
    char Grade;
} SDATA;

SDATA *Head = NULL, **SArray;
int Count = 0;

int main(int argc, char *argv[])
{
    Head = NULL;
    Count = 0;
    ReadInput();
    ConvertToArray();
    CalculateFinalGrade();
    SortByLastName();
    PrintArray("By Last Name");
    SortByFinalScore();
    PrintArray("By Final Score");
    return 0;
}

char First[1000], Last[1000], SSN[20];

void ReadInput(void)
{
    FILE *MyFile;
    int g1, g2, g3;
    SDATA *Temp;

    MyFile = fopen("grades.txt", "r");
    if (MyFile == NULL)
    {
        fprintf(stderr, "Can't open grades.txt\n");
        exit(100);
    }
    // read first line
    fscanf(MyFile, "%s %s %s %d %d %d", Last, First, SSN, &g1, &g2, &g3);
    while (!feof(MyFile))
    {
        Count++;
        Temp = (SDATA *)malloc(sizeof(SDATA));
        if (Temp == NULL)
        {
            fprintf(stderr, "Out of memory\n");
            exit(100);
        }
        Temp->LastName = (char *)malloc(strlen(Last)+1);
        if (Temp->LastName == NULL)
        {
            fprintf(stderr, "Out of memory\n");
            exit(100);
        }
    }
}

```

```

Temp->SocialSecurity = (char *)malloc(strlen(SSN)+1);
if (Temp->SocialSecurity == NULL)
{
    fprintf(stderr,"Out of memory\n");
    exit(100);
}
Temp->FirstName = (char *)malloc(strlen(First)+1);
if (Temp->FirstName == NULL)
{
    fprintf(stderr,"Out of memory\n");
    exit(100);
}
strcpy(Temp->LastName,Last);
strcpy(Temp->FirstName,First);
strcpy(Temp->SocialSecurity,SSN);
Temp->Grade1 = g1;
Temp->Grade2 = g2;
Temp->Grade3 = g3;
// link data structure
Temp->Next = Head;
Head = Temp;
// read next line
fscanf(MyFile,"%s %s %s %d %d %d",Last,First,SSN,&g1,&g2,&g3);
}
fclose(MyFile);
}

void ConvertToArray(void)
{
    SDATA *Temp;
    int i;

    SArray = (SDATA **)calloc(Count,sizeof(SDATA *));
    if (SArray == NULL)
    {
        fprintf(stderr,"Out of memory\n");
        exit(100);
    }
    for (i=0,Temp=Head ; Head ; i++,Temp=Temp->Next)
    {
        SArray[i] = Temp;
    }
}

void CalculateFinalGrade(void)
{
    int i;

    for (i=0 ; i<Count ; i++)
    {
        SArray[i]->Total = (SArray[i]->Grade1+SArray[i]->Grade2+(2*SArray[i]->Grade3))/4;
        if (SArray[i]->Total >= 90)
        {
            SArray[i]->Grade = 'A';
        }
        else if (SArray[i]->Total >= 80)
        {
            SArray[i]->Grade = 'B';
        }
        else if (SArray[i]->Total >= 70)
        {
            SArray[i]->Grade = 'C';
        }
        else if (SArray[i]->Total >= 60)
        {
            SArray[i]->Grade = 'D';
        }
        else
        {
            SArray[i]->Grade = 'F';
        }
    }
}
}

```

